

# Learning to learn STEM courses

Anonymous ACL submission

## Abstract

We curate a new dataset from MIT EECS (Course 6), Physics (Course 8), Economics (Course 14), Mathematics (Course 18), Harvard Statistics, and Columbia Computer Science course questions, transform them into programming tasks using OpenAI Codex, and solve them by executing programs. We curate, transform, and solve ten courses: (i) MIT EECS 6.003 Signal Processing, (ii) MIT EECS 6.036 Introduction to Machine Learning, (iii) MIT EECS 6.042 Mathematics for Computer Science, (iv) MIT Physics 8.282 Introduction to Astronomy, (v) MIT Economics 14.01 Principles of Microeconomics, (vi) MIT Mathematics 18.05 Introduction to Probability and Statistics, (vii) MIT Mathematics 18.06 Linear Algebra, (viii) MIT Mathematics 18.781 Theory of Numbers, (ix) Harvard Statistics STATS110 Probability, and (x) Columbia University COMS3251 Computational Linear Algebra. Our approach works surprisingly well since question solutions and programs share an underlying tree representation. We are able to use Codex to correctly solve all questions by specifying both question and programming contexts such as which mathematical rules to use or which programming packages to load. In addition to generating code which solves problems the resulting code generates plots which are useful for understanding the solutions. We interactively transform the original course questions until they are solved correctly and measure the similarity between the original and transformed questions. Finally, we automatically generate novel questions for each course, providing a way to rapidly synthesize new course content. Our approach is the first scalable solution towards automatically learning to learn all university STEM courses by machine.

## 1 Introduction

Question answering has long been a field of interest in natural language processing (Chen et al., 2017; Rajpurkar et al., 2016; Fader et al., 2013; We-

ston et al., 2015; Richardson et al., 2013; Simmons, 1970; Fader et al., 2014). Large transformer models have successfully performed reading comprehension tasks but often fail at questions involving advanced quantitative reasoning. The successful approaches for solving quantitative problems are usually limited to specific tasks and do not generalize well to out of sample questions in other fields.

Open access question-answer sets from academic courses are useful for evaluating quantitative reasoning questions since a given course subject is likely to have a well-defined scope and difficulty, with questions that are thoughtfully designed specifically to encourage learning and generalization in students. Such question sets may have comprehensive coverage of a range of concepts, and these have often been tested and selected from classes in the past, demonstrating a degree of reliability.

Given the challenge in generalizing from specialized problem solving models, we introduce the application of OpenAI Codex in solving STEM course questions. Codex is a large language model trained on natural language and code. Such foundation models have demonstrated a range of so-called emergent properties, including surprisingly strong performance on new tasks with little to no context (in the form of supervisory examples). We find Codex useful for tackling questions across different STEM courses without fine-tuning per course.

Since quantitative questions and programs alike may be represented as trees or computation graphs, we posit that this results in an overlap in learning to write programs and to solve quantitative problems, supported by tree representations. Additionally, natural language plays a prominent role in both word problems and natural-language-to-code models like Codex, yielding a path to translate questions to programming instructions by paraphrasing.

We transform course questions into programming tasks and successfully solve all questions

044  
045  
046  
047  
048  
049  
050  
051  
052  
053  
054  
055  
056  
057  
058  
059  
060  
061  
062  
063  
064  
065  
066  
067  
068  
069  
070  
071  
072  
073  
074  
075  
076  
077  
078  
079  
080  
081  
082  
083  
084

ID	University	Department	Course	Number
1	MIT	EECS	Signal Processing	6.003
2	MIT	EECS	Introduction to Machine Learning	6.036
3	MIT	EECS	Mathematics for Computer Science	6.042
4	MIT	Physics	Introduction to Astronomy	8.282
5	MIT	Economics	Principles of Microeconomics	14.01
6	MIT	Mathematics	Introduction to Probability and Statistics	18.05
7	MIT	Mathematics	Linear Algebra	18.06
8	MIT	Mathematics	Theory of Numbers	18.781
9	Harvard	Statistics	Probability	STATS110
10	Columbia University	Computer Science	Computational Linear Algebra	COMS3251

Table 1: University STEM courses: we curate, solve, and generate questions for each course.

across ten different university level STEM courses shown in Table 1. These courses reflect different disciplines with different domain heuristics, for example a table of physical constants in astronomy, or conducting simulations in probability and statistics, etc. Our paraphrased programming instructions result in accurate and useful answers to the posed questions, in numerical, symbolic, and pictorial forms. In addition, we report the amount of paraphrasing required per question as a measure for quantifying the work required to transform questions into correctly-answerable tasks, providing benchmarks for future work.

## 2 Related Work

GPT-3 (Brown et al., 2020) is a general purpose language model that shows that language models scale with model size and training for few-shot language tasks. GPT-3 requires tens of thousands of fine-tuned examples to perform well for many tasks. GPT-3 has been used as a pretrained model for many question answering tasks. For reading-comprehension based question answering, GPT-3-based few-shot tuned models perform moderately well on answering multiple-choice questions from humanities courses, but utterly fail at computation-heavy STEM courses. (Hendrycks et al., 2021). The most advanced approach to date, using verifiers with GPT-3 performs below 55% at the high school level on the MATH dataset. (Cobbe et al., 2021). In contrast, our interactive approach solves university level STEM courses perfectly. A transformer based model performs well on solving symbolic integration problems based on symbolic expression input but performs poorly on differential equation problems (Lample and Charton, 2020).

Task-specific models generally perform well, but have not been shown to generalize to other tasks. Graph-embedding tree-decoder approaches perform with at most 88.5% accuracy for high

school math (Xie and Sun, 2020; Zhang et al., 2021). Performance on Machine Learning problems has reached an average of 94% accuracy with a task-specific transformer GNN models (Tran et al., 2021), though this approach works only on the specific course it was trained on and does scale well to many courses.

## 3 Methods

For each of the ten university level STEM courses we curate 20 questions from problem sets, exercises, homeworks, quizzes, and exams. All questions appear in the Appendix. A description of each of the courses is as follows.

**MIT 6.003: Signal Processing.** Fundamentals of signal processing, focusing on the use of Fourier methods to analyze and process signals such as sounds and images. Topics include Fourier series, Fourier transforms, the Discrete Fourier Transform, sampling, convolution, deconvolution, filtering, noise reduction, and compression. Applications draw broadly from areas of contemporary interest with emphasis on both analysis and design. Prereq: 6.0001 and 18.03.

**MIT 6.036: Introduction to Machine Learning** Introduces principles, algorithms, and applications of machine learning from the point of view of modeling and prediction; formulation of learning problems; representation, over-fitting, generalization; clustering, classification, probabilistic modeling; and methods such as support vector machines, hidden Markov models, and neural networks. Prereq: Calculus II and (6.0001 or 6.01).

**MIT 6.042: Mathematics for Computer Science** Elementary discrete mathematics for science and engineering, with a focus on mathematical tools and proof techniques useful in computer science. Topics include logical notation, sets, relations, el-

161	elementary graph theory, state machines and invariants, induction and proofs by contradiction, recurrences, asymptotic notation, elementary analysis of algorithms, elementary number theory and cryptography, permutations and combinations, counting tools, and discrete probability. Prereq: Calculus I.	210
162		211
163		212
164		213
165		214
166		215
167	<b>MIT 8.282: Introduction to Astronomy</b> Quantitative introduction to the physics of planets, stars, galaxies and our universe, from origin to ultimate fate, with emphasis on the physics tools and observational techniques that enable our understanding. Topics include our solar system, extrasolar planets; stellar astronomy, compact objects (white dwarfs, neutron stars, pulsars, black holes); galactic structure, star clusters, interstellar medium, dark matter; quasars, supermassive black holes, gravitational waves; cosmology, gravitational lensing, 21cm tomography. Prereq: Physics.	216
168		217
169		218
170		219
171		220
172		221
173		222
174		223
175		224
176		225
177		226
178		227
179		228
180	<b>MIT 14.01: Principles of Microeconomics</b> Introduces microeconomic concepts and analysis, supply and demand, theories of the firm and individual behavior, competition and monopoly, and welfare economics. Applications to problems of current economic policy. Prereq: None.	229
181		230
182		231
183		232
184		233
185		234
186	<b>MIT 18.05: Introduction to Probability and Statistics</b> Elementary introduction with applications. Basic probability models. Combinatorics. Random variables. Discrete and continuous probability distributions. Statistical estimation and testing. Confidence intervals. Introduction to linear regression. Prereq: Calculus II.	235
187		236
188		237
189		238
190		239
191		240
192		241
193	<b>MIT 18.06: Linear Algebra</b> Basic subject on matrix theory and linear algebra, emphasizing topics useful in other disciplines, including systems of equations, vector spaces, determinants, eigenvalues, singular value decomposition, and positive definite matrices. Applications to least-squares approximations, stability of differential equations, networks, Fourier transforms, and Markov processes. Uses linear algebra software. Compared with 18.700, more emphasis on matrix algorithms and many applications. Prereq: Calculus II.	242
194		243
195		244
196		245
197		246
198		247
199		248
200		249
201		250
202		251
203	<b>MIT 18.781: Theory of Numbers</b> An elementary introduction to number theory with no algebraic prerequisites. Primes, congruences, quadratic reciprocity, diophantine equations, irrational numbers, continued fractions, partitions. Prereq: None.	252
204		253
205		254
206		255
207		256
208	<b>Harvard STAT110: Probability</b> An introduction to probability as a language and set of tools for understanding statistics, science, risk, and randomness. The ideas and methods are useful in statistics, science, engineering, economics, finance, and everyday life. Topics include sample spaces and events, conditioning, Bayes' Theorem. Random variables and their distributions: distributions, moment generating functions, expectation, variance, covariance, correlation, conditional expectation. Univariate distributions and multivariate distributions. Limit theorems, Markov chains: transition probabilities, stationary distributions, reversibility, convergence. Prereq: Calculus.	257
209		258
	<b>Columbia University COMS3251: Computational Linear Algebra</b> An introduction to linear algebra and its usage in computational applications. The study of linear equations, linear functions, and their representations pervades numerous fields of study. Students will learn and practice fundamental ideas of linear algebra and simultaneously be exposed to and work with real-world applications of these ideas. This course emphasizes a rigorous approach to mathematics, which serves as a foundation for future courses like computer graphics, machine learning, and robotics. The learning and usage of Python and libraries such as NumPy is an essential component of the course, as is the development of basic skills of computational programming. Prereq: Calculus.	222
		223
		224
		225
		226
		227
		228
		229
		230
		231
		232
		233
		234
		235
		236
		237
	<b>3.1 Workflows</b>	238
	Figures 1 - 10 illustrate our workflows for solving the ten courses using Codex in diverse ways, showing the original question, the transformed question as input to Codex, the resulting synthesizes program, and executed output answer.	239
		240
		241
		242
		243
	<b>3.2 Experiments</b>	244
	Here we describe the process of using Codex to solve these course problems by (i) transforming each question with contextual information for both the question and program, (ii) using the transformed question as input to Codex, and (iii) executing the output by a Python interpreter to get a correct answer. We describe each of these steps in detail. Key insights from this methodology are in the way paraphrasing the original question is performed. We interactively transform questions until we achieve perfect correct answers on all the problems, and in the process understand and characterize the operations required to obtain these results.	245
		246
		247
		248
		249
		250
		251
		252
		253
		254
		255
		256
		257
		258

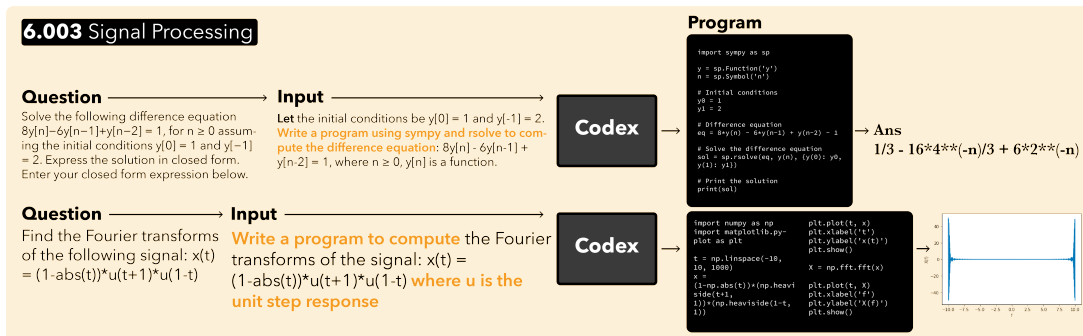


Figure 1: MIT 6.003 Signal Processing workflow: (Top) Codex is prompted to use the symbolic math sympy package to produce code snippets that generate answers in the form of a symbolic mathematical equation. (Bottom) Codex is prompted with contextual information to understand the question correctly, such as the information that  $u$  stands for the unit step response in this context. Given the right input, Codex combines python libraries and uses the matplotlib package to answer the question perfectly along with a plot.

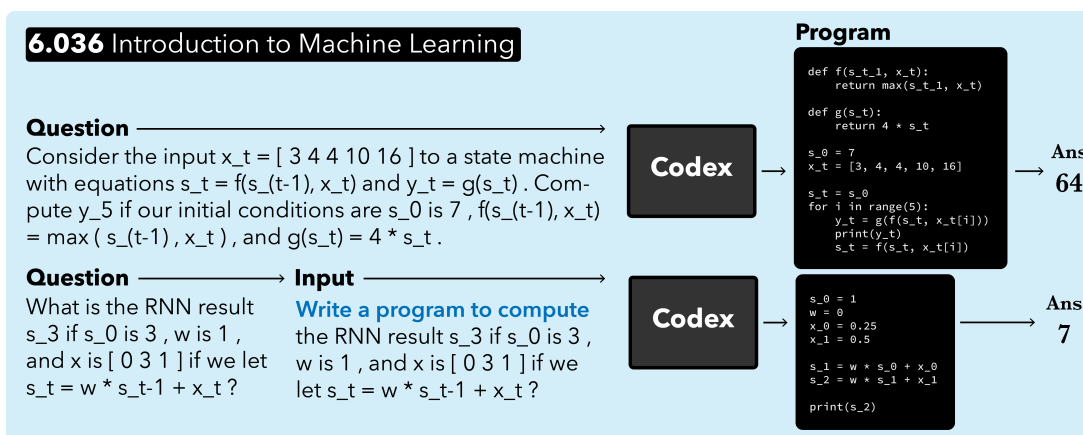


Figure 2: MIT 6.036 Introduction to Machine Learning workflow: the question requires iteratively plugging in values in a recurrent neural network.

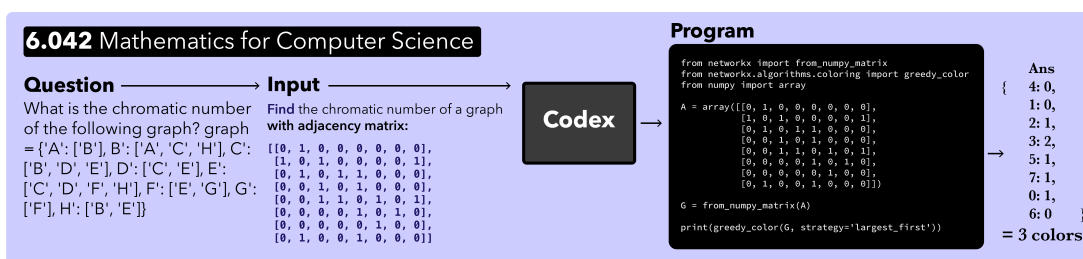


Figure 3: MIT 6.042 Mathematics for Computer Science workflow: In this question we see that the formatting of the graph in the question matters. We also see the solution producing a full coloring of the graph from which the chromatic number may be easily determined.

### 3.3 Evaluation and Question Augmentation

OpenAI's Davinci-Codex is given prompts based on the question. The output is then run using Python 3.6 and is considered correct if the program returns the correct answer or if any printed statement or comment contains the correct answer.

We use Codex with default parameters<sup>1</sup>. If the

<sup>1</sup>Temperature = 0; Responce Length = 200; Top P = 1; Frequency Penalty = 0; Presence Penalty = 0; Best Of = 1

output appears to terminate in the middle of a solution, Codex is allowed to continue to run multiple times. All prompts are surrounded with triple quotation marks, to denote Python comments, as well as new lines.

Often, simply using the exact problem text surrounded by triple quotes is sufficient to obtain a correct answer. For example to solve the question "What is the largest prime factor of 3328?" Codex

259  
260  
261  
262  
263  
264  
265

266  
267  
268  
269  
270  
271  
272  
273  
274



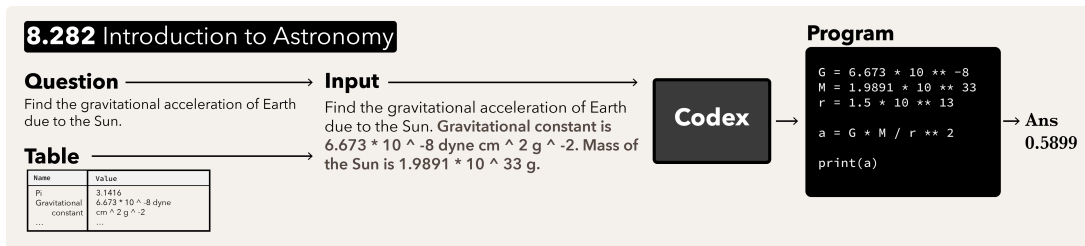


Figure 4: MIT 8.282 Introduction to Astronomy workflow: In this course, Codex often requires context about physical constants. This question involves the gravitational constant (G), mass of the Sun (M), and the distance between the Earth and the Sun (r; this is the definition of one Astronomical Unit).

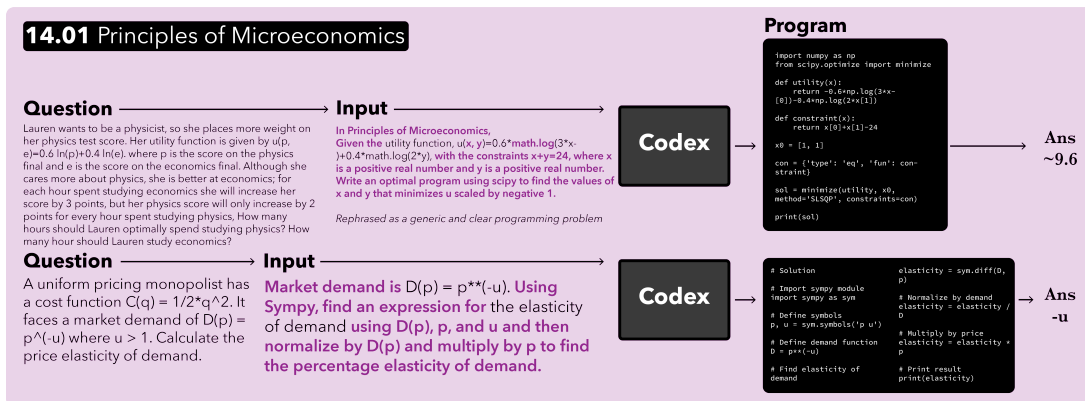


Figure 5: MIT 14.01 Principles of Microeconomics workflow: Codex can be prompted to use the SymPy package to guarantee its attempt at expressing a symbolic solution. Codex can also be prompted to use the SciPy package because SciPy supports non-linear optimization with constraints capability while others do not. Finally, when using the SciPy package, Codex consistently uses the minimize function without adjusting the sign for the objective function, so we must prompt it to flip the sign for the objective function to maximize the utility function.

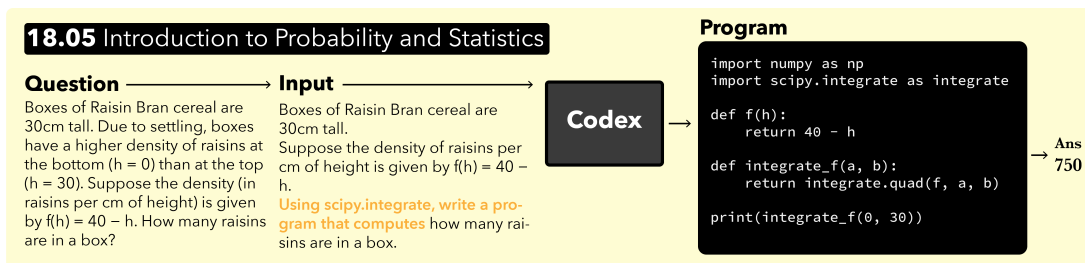


Figure 6: MIT 18.05 Introduction to Probability and Statistics workflow: Codex has to be specifically prompted to use the SciPy package and the integrate function in order to perform the desired task.

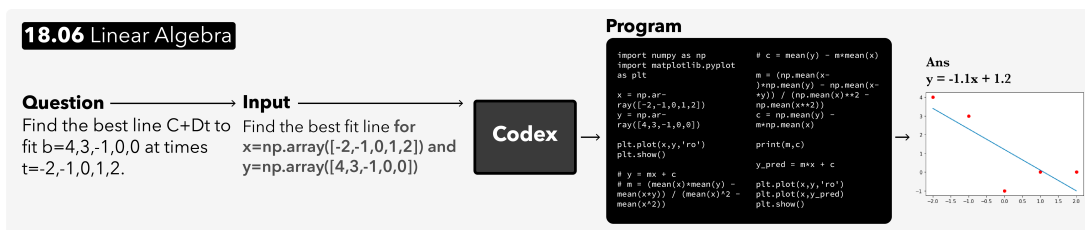


Figure 7: MIT 18.06 Linear Algebra workflow: Codex has the remarkable ability to create figures with multiple equations and datapoints. In this example, inputting the question as is results in a program that not only correctly solves the problem of fitting the best line through the datapoints but also generates the corresponding plots.

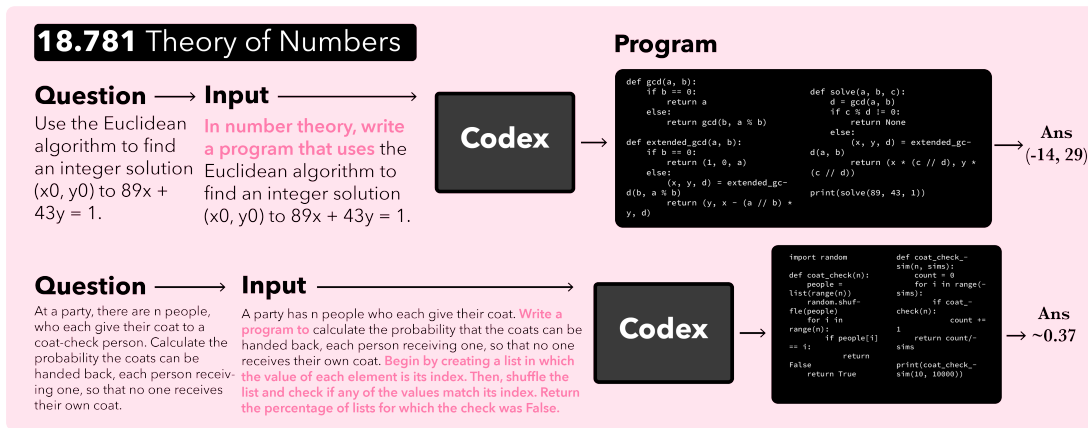


Figure 8: MIT 18.781 Theory of Numbers workflow: (Top) Codex successfully solves a problem with a prompt nearly identical to the original problem. All that was changed was a brief hint suggesting that the problem belongs to the topic of number theory, and that the solution should take the form of a program. (Bottom) Codex requires a highly explicit problem definition in order to solve the problem, it approximates the solution with a Monte Carlo method. Without the final sentence reiterating that the probability should correspond to the likelihood that no one receives their coat back, Codex gives a solution that solves for the unlikelihood instead.

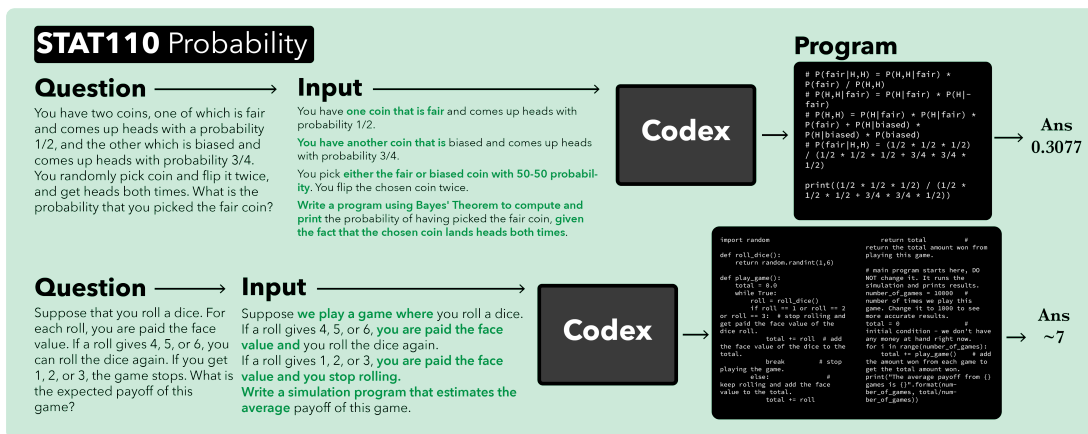


Figure 9: Harvard STAT110 Probability workflow: (Top) Codex requires separation of key question details into separate statements that avoid language ambiguities. The last sentence indicates the Bayesian nature of the problem more explicitly. (Bottom) In some cases, codex requires usage of programming specific concepts to guide solution generation. Additionally, sequence transition words guide the solution towards performing tasks in a desired order.

tests whether a number divides 3328 and is prime. However, Codex is sensitive to input formats and the choice of language for the question. In addition, questions may be messy: missing operators, values, definitions, or vaguely worded. We therefore allow variations of the questions as a cleaning process to obtain a correct answer by modifying the exact original question wording. The process of engineering these prompts is performed interactively.

Most problems are solved with minor augmentations to the wording of the question as well as simplifying, standardizing, adding a missing definition, or often cleaning by removing ‘noisy’ elements. Synonyms for words are used, such as ‘calculate’ versus ‘outputs’ versus ‘gives’, or ‘shows’ versus

‘sample’ versus ‘value’. We may add guidance to use parenthesis, and may replace blanks with multiplication operators if appropriate. Further, if the program calls an undefined function that did not appear to be part of the standard library, we prompt the program to import python libraries. We continue until any of these produce the correct answer.

For questions which require more substantive changes to solve the problem, related information such as the type of problem and associated definitions may be included in the problem. For example, we may add the information that the problem is about complex numbers. Story elements may be changed or stripped away. A question with multiple

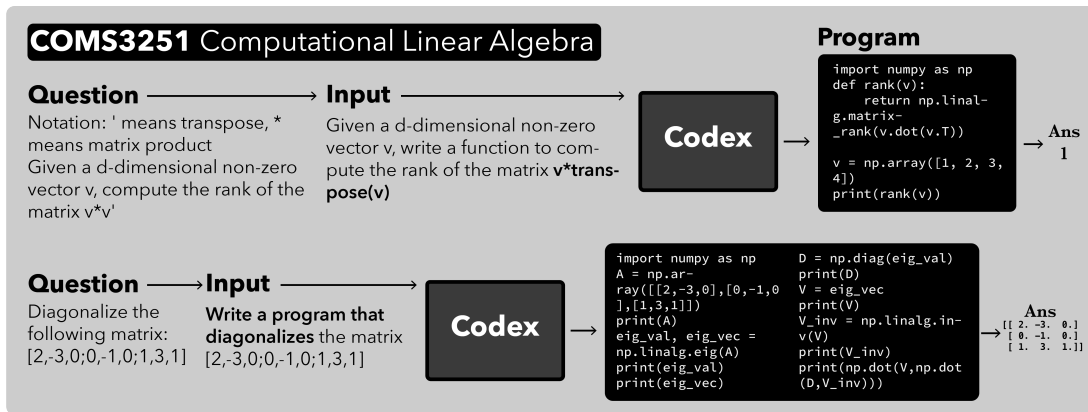


Figure 10: COMS3251 Computational Linear Algebra workflow: (Top) To avoid confusion over the usage of  $\cdot$ , the transpose operation is written as a programming method. Codex generates a general-purpose rank function that also generalizes to other problems. (Bottom) Codex is prompted to "write a program" to indicate preference towards the main sequential program instead of new function definitions.

parts may be split into multiple programming tasks. Additional tasks may also be used to help generate the correct output. For example, we may add a task to plot and illustrate parts of the problem solution. The ability to add information as well as augment the answer with plots is useful for understanding problem solutions.

### 3.4 Successful Augmentation Techniques

A useful technique is to provide Codex context about the problem or program required to solve it. For example, adding a sub-topic or class name before the question. In another case we provide context that the problem is about complex numbers. Many questions implicitly assume certain domains such as integers, reals, complex numbers, booleans, or finite fields, and this may be confusing when the question is taken out of its original context.

Similar to adding context to the problem it may be useful to add context to the program by prompting Codex to use certain Python libraries such as sympy a symbolic math library, numpy a numerical math library, and itertools which deals with permuting and iterating through lists. This provides context on the type of problem involved. For example problems involving manipulating polynomials are often solved with sympy. Another useful technique is breaking tasks into simple steps, breaking complex sentences into simple ones. Many equations may be unclear, ambiguous, or use notation specialized to a certain subject. For example, a question may imply multiplication without using a multiplication sign or may not place parentheses around arguments to trigonometric functions. An-

other example is using  $i$  or  $j$  for complex numbers or  $\mathbb{R}$  versus  $\mathbb{R}$  to denote the set of real numbers. Even symbols that appear the same to a human may have different encodings. Varying these style choices, especially by making them less ambiguous and closer to notation found in Python programming seems to make a substantial difference.

## 4 Results

### 4.1 Measuring Question Similarity

As a way to quantify how much a question is altered before reaching the correct answer, we use a measure of similarity based on the cosine angle difference between the sentence embeddings of the original question and the Codex prompt. A plot of the similarity between the original question and the prompt given to Codex which produced a correct answer is shown in Figure 11. The baseline is calculated from the mean of the pairwise differences between all of the individual questions in the topic.

### 4.2 Generating New Questions

We use the curated questions to generate new question for each of the courses. Examples can be seen in Table 2.

## 5 Conclusion

This work introduces a simple and scalable, yet powerful, technique for solving quantitative STEM course problems. We find that these quantitative questions often have an underlying expression tree which solves the question. Analogously, code programs also have an underlying expression tree representation. By converting these course questions

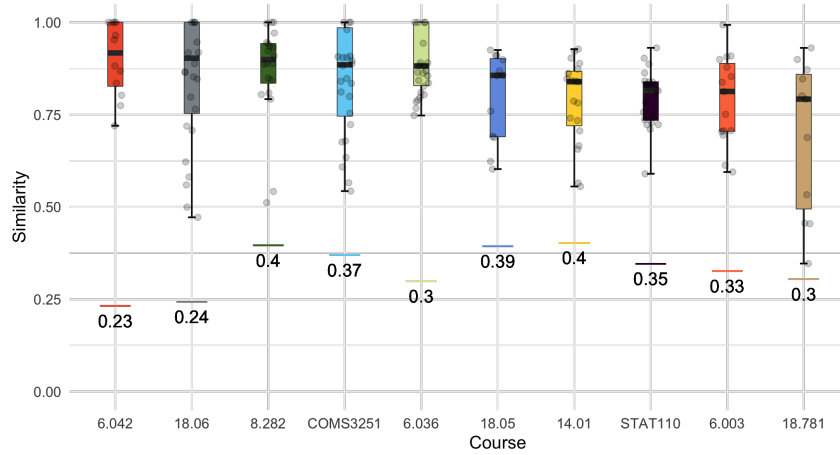


Figure 11: Similarity between original questions and final engineered prompts by course. We hypothesize that less difficult courses should need less substantial modifications to their questions in order for Codex to generate a successful answering program. Here, courses are ordered by this measure (median across questions). The lines below represent "baseline" similarity numbers, for reference. These are computed by taking a sample of question pairs within a course (e.g. Q1 and Q10), then computing their similarity, and then averaging across all such similarities within a given course.

ID	Course	Automatically generated question
1	MIT 6.003	Find the Fourier transforms of the following signal: $x(t) = \exp(-\text{abs}(t)) * \cos(2 * t)$
2	MIT 6.003	Use the initial and final value theorems (where applicable) to find $x(0)$ and $x(\text{inf})$ for the signal with the following Laplace transform: $1/(s^2 + 1)$
3	MIT 6.036	Consider the following RNN: $s_t = w * s_{t-1} + x_t$ , where $s_0$ is 1, $w$ is 0, and $x$ is [ 0.25 0.5 ]. What is the output of the RNN if we input [ 0.25 0.5 ] ?
4	MIT 6.036	Consider the following neural network with one hidden layer of size 3. The input is [ 1 2 3 ] and the output is [ 1 2 3 ]. The weights are [ 1 1 1 ; 1 1 1 ; 1 1 1 ] and [ 1 1 1 ]. What is the output of the network if we apply the ReLU activation function to the hidden layer and the sigmoid activation function to the output layer?
5	MIT 6.042	Let $X$ be a random variable with values in the set 1, 2, 3, 4, 5, 6. Find $E[X]$ and $Var(X)$ .
6	MIT 6.042	A standard die is rolled until the first time a six is rolled. What is the probability that the number of rolls is less than or equal to 3?
7	MIT 8.282	A neutron star has a mass of $1.4 M_{\odot}$ , a radius of 10 km, and a moment of inertia of $I = 10^{45} \text{ g/cm}^2$ . What is the minimum rotation period of this star?
8	MIT 8.282	A star is observed to have a period of $P = 0.5$ days, and a radius of $R = 0.1 R_{Sun}$ . What is the mass of the star?
9	MIT 14.01	A monopolist faces the following demand curve: $Q_D(P) = -2 + 0.5 * P$ . The monopolist has a cost function $C(q) = q^2 + q + 2$ . Plot the profit function and the marginal revenue function.
10	MIT 14.01	Consider the following production function: $F(L, K) = L^{1/3} + K^{1/3}$ . (a) If $L = 1$ and $K = 1$ , what is the marginal product of labor? (b) If $L = 1$ and $K = 1$ , what is the marginal product of capital?
11	MIT 18.05	Suppose that the average height of a male in a population is 70 inches with a standard deviation of 4 inches. What is the probability that a male selected at random is at least 72 inches tall?
12	MIT 18.05	Calculate the probability of getting a full house poker hand.
13	MIT 18.06	Find the determinant of A, B, C : A = [0,4;3,0], B = [2,0;4,2], and C = [3,4;5,7].
14	MIT 18.06	Find the eigenvalues and eigenvectors of A, B, C : A = [0,4;3,0], B = [2,0;4,2], and C = [3,4;5,7].
15	MIT 18.781	Find the number of integers between 1 and $10^6$ that are relatively prime to $10^{12}$ .
16	MIT 18.781	Find the smallest prime $p$ such that the polynomial $x^2 - x - 1$ is irreducible over $\text{GF}(p)$ .
17	COMS3251	Find the eigenvalues of the following matrix: [1,2,3;2,4,5;3,5,6]
18	COMS3251	Compute the determinant of the following matrix: [1,2,3;4,5,6;7,8,9]
19	STAT110	You have a basket of $n$ fruits. Each fruit has a color, and each color has an equal probability of $3/5$ . You randomly take a fruit from the basket, then you randomly take a fruit from the basket again without putting the first fruit back in. What is the probability that the two fruits have the same color?
20	STAT110	A certain couple tells you that they have two children, at least one of which is a girl. What is the probability that they have two girls?

Table 2: New questions generated from all ten courses.

to tasks for program synthesis and using Codex to generate programs which return the solution, we are able to solve even relatively challenging questions across a variety of courses which each have domain characteristics and expected output formats. We demonstrate methods for modifying questions into such formats for successfully obtaining correct answers, and quantify the degree to which differ-

ent courses' questions need explicit modification. Our work is a step towards fully automatic paraphraser, based on our data and approaches, that may be applied to large-scale open-domain quantitative reasoning tasks with high quality, steerable, and interpretable results. Our work has useful applications in education, including tutoring, automated grading, and automated content generation.



387  
388  
389  
390  
391  
392  
  
393  
394  
395  
  
396  
397  
398  
399  
  
400  
401  
402  
403  
404  
  
405  
406  
407  
408  
409  
  
410  
411  
412  
413  
414  
  
415  
416  
417  
  
418  
419  
420  
421  
  
422  
423  
424  
425  
426  
427  
  
428  
429  
430  
  
431  
432  
433  
434  
  
435  
436  
437  
438  
439

## References

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv:2110.14168*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2014. Open question answering over curated and extracted knowledge bases. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1156–1165.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *International Conference on Learning Representations*.

Guillaume Lample and François Charton. 2020. Deep learning for symbolic mathematics. *International Conference on Learning Representations*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 193–203.

Robert F Simmons. 1970. Natural language question-answering systems: 1969. *Communications of the ACM*, 13(1):15–30.

Sunny Tran, Pranav Krishna, Ishan Pakuwal, Prabhakar Kafle, Nikhil Singh, Jayson Lynch, and Iddo Drori. 2021. Solving machine learning problems. *Asian Conference on Machine Learning*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*.

Zhipeng Xie and Shichao Sun. 2020. A goal-driven tree-structured neural model for math word problems. *International Joint Conference on Artificial Intelligence*. 440  
441  
442  
443

Jipeng Zhang, Lei Wang, Roy Ka-Wei Lee, Yi Bin, Yan Wang, Jie Shao, and Ee-Peng Lim. 2021. Graph-to-tree learning for solving math word problems. *Asian Conference on Machine Learning*. 444  
445  
446  
447

## A MIT 6.003: Signal Processing

Topic	Signals and systems, Discrete-time systems
Original Question	Solve the following differential equation $y(t) + 3dy(t)/dt + 2d^2y(t)/dt^2 = 1$ for $t \geq 0$ assuming the initial conditions $y(0) = 1$ and $dy(t)/dt = 2$ when $t = 0$ . Express the solution in closed form. Enter your closed form expression below.
Codex Input	Let the initial conditions be $y(0) = 1$ and $dy(t)/dt = 2$ when $t=0$ . Write a program using sympy and dsolve to compute the differential equation: $y(t) + 3dy(t)/dt + 2d^2y(t)/dt^2 = 1$ , where $t \geq 0$ .
Codex Output	<pre>import sympy as sp  t = sp.symbols('t') y = sp.Function('y')  # Initial conditions ics = {y(0): 1, y(t).diff(t).subs(t, 0): 2}  # The differential equation eq = y(t) + 3*y(t).diff(t) + 2*y(t).diff(t, t) - 1  # Solve the differential equation sol = sp.dsolve(eq, y(t), ics=ics)  print(sol)</pre>
Solution	$y(t) = -4 * \exp(-t) + 4 * \exp(-t/2) + 1$

### Question 1 from MIT 6.003

Topic	Signals and systems, Discrete-time systems
Original Question	Solve the following difference equation $8y[n]6y[n1] + y[n2] = 1$ , for $n \geq 0$ assuming the initial conditions $y[0] = 1$ and $y[1] = 2$ . Express the solution in closed form. Enter your closed form expression below.
Codex Input	Let the initial conditions be $y[0] = 1$ and $y[-1] = 2$ . Write a program using sympy and rsolve to compute the difference equation: $8y[n] - 6y[n - 1] + y[n - 2] = 1$ , where $n \geq 0$ , $y[n]$ is a function.
Codex Output	<pre>import sympy as sp  y = sp.Function('y') n = sp.Symbol('n')  # Initial conditions y0 = 1 y1 = 2  # Difference equation eq = 8*y(n) - 6*y(n-1) + y(n-2) - 1  # Solve the difference equation sol = sp.rsolve(eq, y(n), {y(0): y0, y(1): y1})  # Print the solution print(sol)</pre>
Solution	$y[n] = 1/6 * (1/4)^n + 1/2 * (1/2)^n + 1/3$

### Question 2 from MIT 6.003

Topic	Signals and systems, Discrete-time systems
Original Question	Expand $1/(1-a)$ in a power series.
Codex Input	Write a program to expand $1/(1-a)$ into a power series.
Codex Output	<pre> from sympy import *  x = Symbol('x') a = Symbol('a') f = 1/(1-a) print(fps(f))  from sympy import * from fps import *  def main():     a = Symbol('a')      # 1/(1-a) = 1 + a + a^2 + a^3 + ...      # expand 1/(1-a) into a power series and write the first     # four terms     print(fps(1/(1-a), 4)) </pre>
Solution	$1 + a + a^2 + a^3 + \dots$

Question 3 from MIT 6.003

Topic	Signals and systems, Discrete-time systems
Original Question	Expand $1/(1-a)^2$ in a power series.
Codex Input	Using sympy and fps, write a program to expand $1/(1-x)^2$ into a power series
Codex Output	<pre> from sympy import fps, symbols, init_printing init_printing() x = symbols('x') fps(1/(1-x)**2).truncate(10) </pre>
Solution	$1 + 2a + 3a^2 + 4a^3 + \dots$

Question 4 from MIT 6.003

Topic	Feedback, poles, and fundamental modes, Continuous-time systems
Original Question	Let $h[n]$ represent the $n$ th sample of the unit-sample response of a system with system functional $H(R)$ . Determine $h[2]$ of the following system: $H[R] = R/(1 - (3/4) * R)$
Codex Input	In signal processing, using sympy fps write a program to compute the second sample of the unit-sample response of the system $H[x] = x/(1 - (3/4) * x)$
Codex Output	<pre> from sympy import fps, symbols, sympify, init_printing, pprint  init_printing()  x = symbols('x')  H = x/(1-(3/4)*x)  fps(H).truncate() </pre>
Solution	$h[2] = 3/4$

Question 5 from MIT 6.003

Topic	Feedback, poles, and fundamental modes, Continuous-time systems
Original Question	Let $h[n]$ represent the $n$ th sample of the unit-sample response of a system with system functional $H(R)$ . Determine $h[2]$ of the following system: $H[R] = 1/(1 - R)^2$
Codex Input	In signal processing, using sympy fps write a program to compute the second sample of unit-sample response of the system $H[x] = 1/((1 - x) ** 2)$ .
Codex Output	<pre> from sympy import fps, symbols, sympify, init_printing,     pprint  init_printing()  x = symbols('x')  H = 1 / ((1-x)**2)  fps(H).truncate()</pre>
Solution	$h[2] = 3$

Question 6 from MIT 6.003

Topic	Feedback, poles, and fundamental modes, Continuous-time systems
Original Question	Let $h[n]$ represent the $n$ th sample of the unit-sample response of a system with system functional $H(R)$ . Determine $h[2]$ of the following system: $H[R] = 1/((1 - 0.5 * R) * (1 - 0.25 * R))$
Codex Input	In signal processing, using sympy fps write a program that computes the second sample of the unit-sample response of the system $H[R] = 1/((1 - 0.5 * R) * (1 - 0.25 * R))$
Codex Output	<pre> from sympy import fps, symbols, sympify, init_printing,     pprint  init_printing()  R = symbols('R')  H = 1 / ((1 - 0.5 * R) * (1 - 0.25 * R))  fps_H = fps(H)  pprint(fps_H)  pprint(fps_H.truncate(n=2))</pre>
Solution	$h[2] = 7/16$

Question 7 from MIT 6.003



Topic	Feedback, poles, and fundamental modes, Continuous-time systems
Original Question	Consider a system H whose unit-sample response is $h[n] = n + 1$ for $n \geq 0$ , $h[n] = 0$ otherwise, determine the poles of H.
Codex Input	In signal processing, write a program to find the system's pole, whose unit-sample response is $h[n] = n + 1$ for $n \geq 0$ , and $h[n] = 0$ otherwise.
Codex Output	<pre>import numpy as np import matplotlib.pyplot as plt  def unit_sample_response(n):     return n+1  def pole_finder(n):     return np.exp(1j*2*np.pi*n)  def plot_unit_sample_response(n):     plt.plot(n, unit_sample_response(n))     plt.show()  def plot_pole_finder(n):     plt.plot(n, np.abs(pole_finder(n)))     plt.show()  if __name__ == "__main__":     n = np.arange(0, 100)     plot_unit_sample_response(n)     plot_pole_finder(n)</pre>
Solution	two poles, each at $z = 1$

#### Question 8 from MIT 6.003

Topic	Z transform, Laplace transform
Original Question	Evaluate the real part of $j^j$
Codex Input	Write a program to evaluate the real part of $j^j$
Codex Output	<pre>import numpy as np  def jj(n):     j = np.complex(0, 1)     return np.power(j, j)  print(jj(1))</pre>
Solution	$exp(-pi/2)$

#### Question 9 from MIT 6.003

Topic	6.003 Signals and Systems
Original Question	Evaluate the real part of $(1 - j * \sqrt{3})^{12}$
Codex Input	Write a program to evaluate the real part of $(1 - j * \sqrt{3})^{12}$
Codex Output	<pre>import cmath  def main():     """     program to evaluate the real part of (1 - j*sqrt(3))^12     """     num = (1 - 1j*cmath.sqrt(3))     num = num**12     print(num.real)  if __name__ == "__main__":     main()</pre>
Solution	4096

#### Question 10 from MIT 6.003

Topic	Z transform, Laplace transform
Original Question	Determine the Z transform of the following signal: $x[n] = (1/2)^n * u[n3]$
Codex Input	In signal processing, write a program to perform the Z transform on the signal: $x[n] = (1/2)^n * u[n3]$ where $u$ is the unit time response.
Codex Output	<pre>import numpy as np import matplotlib.pyplot as plt  def z_transform(x):     z = np.exp(1j*2*np.pi/len(x)*np.arange(len(x)))     return np.array([np.sum(x*np.power(0.5,n)*z**(-n)) for n                       in range(len(x))])  x = np.array([(1/2)**n*(1 if n&gt;=0 and n&lt;3 else 0) for n in               range(10)])  plt.plot(np.real(z_transform(x))) plt.show()</pre>
Solution	$X = 1/(8 * z^2 * (z - 1/2))$

Question 11 from MIT 6.003

Topic	Z transform, Laplace transform
Original Question	If the region of convergence of a system's Z transform includes the unit circle, is the system stable?
Codex Input	In signal processing, write a program to compute if a system is stable where the region of convergence in a system's Z transform includes the unit circle.
Codex Output	<pre># Z transform of a stable system is a circle. # The region of convergence is the unit circle. # Therefore, the system is stable.</pre>
Solution	Yes

Question 12 from MIT 6.003

Topic	Z transform, Laplace transform
Original Question	Consider this variant of the Fibonacci system: $y[n] = y[n1]y[n2] + x[n]$ , compute the period of its unit-sample response.
Codex Input	In signal processing, write a program to compute the unit-sample response of the system: $y[n] = y[n1]y[n2] + x[n]$
Codex Output	<pre>import numpy as np import matplotlib.pyplot as plt  x = np.array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1]) y = np.zeros(10)  for i in range(2, 10):     y[i] = y[i - 1] - y[i - 2] + x[i]  plt.stem(y) plt.show()</pre>
Solution	6

Question 13 from MIT 6.003

Topic	Z transform, Laplace transform
Original Question	Consider this variant of the Fibonacci system: $y[n] = y[n-1]y[n-2] + x[n]$ where $x[n]$ represents the input and $y[n]$ represents the output. Enter the poles of the system.
Codex Input	The difference equation $y[n] = y[n-1]y[n-2] + x[n]$ . Use the transfer function and scipy to find the poles of the system.
Codex Output	<pre>import numpy as np import matplotlib.pyplot as plt from scipy import signal  # Transfer function: H(z) = 1 / (z^2 - z + 0.5) b, a = [1, -1, 0.5], [1, 0, 0] w, h = signal.freqz(b, a)  plt.plot(w, 20 * np.log10(abs(h))) plt.xscale('log') plt.title('Chebyshev Type II frequency response (rs=40)') plt.xlabel('Frequency [radians / second]') plt.ylabel('Amplitude [dB]') plt.margins(0, 0.1) plt.grid(which='both', axis='both') plt.axvline(100, color='green') # cutoff frequency plt.show()  # Poles print(signal.tf2zpk(b, a))  # zeros print(signal.tf2zpk([1], [1, 0, 0.5])[0])</pre>
Solution	$poles = e^{j\pi/3}e^{-j\pi/3}$

Question 14 from MIT 6.003

Topic	Discrete approximation of continuous-time systems, Convolution
Original Question	Determine the Laplace transforms (including the regions of convergence) of the following signal: $x(t) = \exp(2 * (t3))u(t3)$
Codex Input	Use sympy to write a program that computes the Laplace transform of the signal $x(t) = \exp(2 * (t3)) * u(t3)$
Codex Output	<pre>from sympy import * from sympy.abc import t, s, a from sympy.integrals.transforms import     inverse_laplace_transform  x = exp(-2*(t-3))*Heaviside(t-3) X = laplace_transform(x, t, s) print(X[0])</pre>
Solution	$X = \exp(-3 * s)/(s + 2)$

Question 15 from MIT 6.003

Topic	Discrete approximation of continuous-time systems, Convolution
Original Question	Determine the Laplace transforms (including the regions of convergence) of the following signal: $x(t) = (1 - (1 - t) * \exp(-3 * t))u(t)$
Codex Input	Use sympy to write a program that computes the Laplace transform of the signal $x(t) = (1 - (1 - t) * \exp^{-3*t}) * u(t)$
Codex Output	<pre> from sympy import * from sympy.integrals import laplace_transform from sympy.abc import t, s, a import matplotlib.pyplot as plt import numpy as np  # Laplace transform of the signal x(t) = # (1 - (1-t)*exp^(-3*t))*u(t) x = (1 - (1-t)*exp(-3*t))*Heaviside(t) X = laplace_transform(x, t, s)[0] print('X(s)=', X) </pre>
Solution	$X = (4 * s + 9) / (s * (s + 3)^2)$

Question 16 from MIT 6.003

Topic	Discrete approximation of continuous-time systems, Convolution
Original Question	Use the initial and final value theorems (where applicable) to find $x(0)$ and $x(\infty)$ for the signal with the following Laplace transform: $1 / (s * (s + 1)^2)$
Codex Input	Use the inverse_laplace_transform to find original function. The resulting laplace transform: $1 / (s * (s + 1)^2)$ . Use a laplace transform to find $x(0)$ and $x(\infty)$ for the signal.
Codex Output	<pre> from sympy import * from sympy.abc import s, t from sympy.integrals import inverse_laplace_transform  # Original function f = Function('f')  # Laplace transform F = 1 / (s*(s+1)**2)  # Inverse laplace transform f = inverse_laplace_transform(F, s, t)  # x(0) x_0 = f.subs(t, 0)  # x(inf) x_inf = limit(f, t, oo)  print(f'Original function: {f}') print(f'x(0): {x_0}') print(f'x(inf): {x_inf}') </pre>
Solution	$x(0) = 0, x(\infty) = 1$

Question 17 from MIT 6.003



Topic	Discrete approximation of continuous-time systems, Convolution
Original Question	Use the initial and final value theorems (where applicable) to find $x(0)$ and $x(\infty)$ for the signal with the following Laplace transform: $1/(s * (s + 1)^2)$
Codex Input	Use the inverse_laplace_transform to find original function. The resulting laplace transform: $1/(s * (s + 1)^2)$ . Use a laplace transform to find $x(0)$ and $x(\infty)$ for the signal.
Codex Output	<pre> from sympy import * from sympy.abc import s, t from sympy.integrals import inverse_laplace_transform  # Original function f = Function('f')  # Laplace transform F = 1/(s*(s+1)**2)  # Inverse laplace transform f = inverse_laplace_transform(F, s, t)  # x(0) x_0 = f.subs(t, 0)  # x(inf) x_inf = limit(f, t, oo)  print(f'Original function: {f}') print(f'x(0): {x_0}') print(f'x(inf): {x_inf}') </pre>
Solution	$x(0) = 0, x(\infty) = 1$

Question 18 from MIT 6.003

Topic	Discrete approximation of continuous-time systems, Convolution
Original Question	Use the initial value theorems (where applicable) to find $x(0)$ for the signal with the following Laplace transform: $1/(s^2 + 1)$
Codex Input	In signal processing, write a program using both laplace_transform and inverse_laplace_transform to find $x(0)$ for the signal with the laplace transform: $1/(s^2 + 1)$ . Use the initial value theorem.
Codex Output	<pre> from sympy import * from sympy.abc import s,t  x = Function('x')  f = 1/(s**2+1)  print(f)  F = laplace_transform(f, s, t)  print(F) </pre>
Solution	$x(0) = 0$

Question 19 from MIT 6.003

Topic	Topic: Feedback and control, Continuous-time frequency response and Bode plots
Original Question	For system $1/(1 + s + s^2)$ , find the frequency for which the magnitude of the gain is greatest.
Codex Input	In signal processing, write a program to find the frequency which maximizes the magnitude of the gain in the system $1/(1+s+s^2)$ .
Codex Output	<pre>import numpy as np import matplotlib.pyplot as plt  def freq_max_mag(w):     return 1/(1+w+w**2)  w = np.linspace(-10,10,1000)  plt.plot(w, freq_max_mag(w)) plt.show()</pre>
Solution	$\sqrt{1/2}$

Question 20 from MIT 6.003

## B MIT 6.036: Introduction to Machine Learning

449

Topic	Basics
Original Question	Let an input vector be [ 5 1 1 ]. What is its magnitude?
Codex Input	Let an input vector be [ 5 1 1 ]. What is its magnitude?
Codex Output	<pre>import math  def magnitude(x):     return math.sqrt(sum(i**2 for i in x))  x = [5,1,1] print(magnitude(x))</pre>
Solution	5.196152423

### Question 1 from MIT 6.036

Topic	Basics
Original Question	If $x = [4\ 4]$ , what is $\ x\ $ ?
Codex Input	If $x = [4\ 4]$ , what is $\ x\ $ ?
Codex Output	<pre>def norm(x):     return np.sqrt(np.sum(x**2))  x = np.array([4, 4]) print(norm(x))</pre>
Solution	5.656854249

### Question 2 from MIT 6.036

Topic	Basics
Original Question	Find the Euclidean length of [ 1 4 3 ] .
Codex Input	Find the Euclidean length of [ 1 4 3 ] .
Codex Output	<pre>import numpy as np  a = np.array([1, 4, 3])  print(np.linalg.norm(a))</pre>
Solution	5.099019514

### Question 3 from MIT 6.036

Topic	Perceptrons
Original Question	Calculate the maximum number of possible mistakes made by the perceptron algorithm if the margin of the separator is 4 and the maximum magnitude of a point is 8 .
Codex Input	Calculate the maximum number of possible mistakes made by the perceptron algorithm if the margin of the separator is 4 and the maximum magnitude of a point is 8 .
Codex Output	<pre>def max_mistakes(margin, max_magnitude):     return (max_magnitude * margin) ** 2  print(max_mistakes(4, 8))</pre>
Solution	1024

### Question 4 from MIT 6.036

Topic	Perceptrons
Original Question	Consider the classifier [ 1 3 0 ] and [ 2 3 1 ] . Do they represent the same classifier ? Return 1 if true and another value if false .
Codex Input	Consider the classifier [ 1 3 0 ] and [ 2 3 1 ] . Do they represent the same classifier ? Return 1 if true and another value if false .
Codex Output	<pre>def same_classifier(v1, v2):     if v1[0] == v2[0] and v1[1] == v2[1] and v1[2] == v2[2]:         return 1     else:         return 0</pre>
Solution	4

Question 5 from MIT 6.036

Topic	Perceptrons
Original Question	A classifier has a decision boundary where theta is ( 1 3 ) . What value does it classify p , where p is ( 0 negative 1 ) ?
Codex Input	A classifier has a decision boundary where theta is ( 1 3 ) . Write a program to determine the explicit value it classifies p , where p is ( 0 -1 ) . Take the dot product between theta and p.
Codex Output	<pre>import numpy as np  classifier = np.array([1, 3]) point = np.array([0, -1])  print(np.dot(classifier, point))</pre>
Solution	-3

Question 6 from MIT 6.036

Topic	Features
Original Question	Given the values for theta as 2 and theta_0 as 1 , compute the NLL loss on the data point ( 2 2 ) . Use log base e of 2.71828 for the log .
Codex Input	Given the values for theta as 2 and theta_0 as 1 , write a program that computes the negative log likelihood loss on the data point ( 2 2 ) . Use log base e of 2.71828 for the log . Negate the result.
Codex Output	<pre>import numpy as np  def loss(x, y, theta, theta_0):     return -(y * np.log(sigmoid(x, theta, theta_0)) + (1 -         y) * np.log(1 - sigmoid(x, theta, theta_0)))  def sigmoid(x, theta, theta_0):     return 1 / (1 + np.exp(-theta * x - theta_0))  print(loss(2, 2, 2, 1)) print(loss(2, 2, 2, 1) * -1)</pre>
Solution	4.993284624

Question 7 from MIT 6.036



Topic	Features
Original Question	What is the result of applying the value negative 2 to the sigmoid function ? Let e be equal to 2.71828 .
Codex Input	Write a program to compute the result of applying the value negative 2 to the sigmoid function ? Let e be equal to 2.71828.
Codex Output	<pre>import math def sigmoid(x):     return 1 / (1 + math.exp(-x))  print(sigmoid(-2))</pre>
Solution	0.1192030633

Question 8 from MIT 6.036

Topic	Features
Original Question	What is the size of the margin of a point 1 by a classifier with theta 1 and theta_0 negative 2 if the point has label negative 1 ?
Codex Input	What is the size of the margin of a point 1 by a classifier with theta 1 and theta_0 negative 2 if the point has label negative 1 ?
Codex Output	<pre>def margin_size(x, theta, theta_0, label):     return label * (np.dot(theta, x) - theta_0)  print(margin_size(1, 1, -2, -1))</pre>
Solution	-3

Question 9 from MIT 6.036

Topic	Logistic Regression
Original Question	Calculate the value of the function $(2 * \theta + \text{negative } 2)^4$ after updating the theta value in one step of gradient descent . Have theta be 4 and eta be 0.01 .
Codex Input	Calculate the value of the function $(2 * \theta + \text{negative } 2)^4$ after updating the theta value in one step of gradient descent . Have theta be 4 and eta be 0.01 .
Codex Output	<pre>import sympy as sp  theta = sp.symbols('theta')  f = (2 * theta - 2) ** 4  f_prime = f.diff(theta)  f_prime_value = f_prime.subs(theta, 4)  print(f_prime_value)  theta_new = theta - 0.01 * f_prime_value  print(theta_new)  f_new = f.subs(theta, theta_new)  print(f_new)</pre>
Solution	665323.4217

Question 10 from MIT 6.036

Topic	Logistic Regression
Original Question	Let theta be ( negative 2 1 ) , theta_0 be 0.25, and x be ( 0 , negative 1 ) . Compute theta times x plus theta_0 .
Codex Input	Let theta be ( negative 2 1 ) , theta_0 be 0.25, and x be ( 0 , negative 1 ) . Compute theta times x plus theta_0 .
Codex Output	<pre># Solution  # Type your code here theta = np.array([-2,1]) theta_0 = 0.25 x = np.array([0,-1]) print(np.dot(theta,x)+theta_0)</pre>
Solution	-0.75

Question 11 from MIT 6.036

Topic	Neural Networks I
Original Question	A neural network has inputs x1 = 1 with weight 2 and x2 = 4 with weight 1 and offset value oA = 0.5 . Neuron B inputs x2 with offset 1 . Neuron C takes in the output of neurons A and B with offsets wAC = 1 and wBC = 3 , respectively . Neuron C has offset value oC = 5 . Compute the output .
Codex Input	Make a machine learning neural network. A neural network has inputs x1 = 1 with weight 2 and x2 = 4 with weight 1 and offset value oA = 0.5 . Neuron B inputs x2 with offset 1 . Neuron C takes in the output of neurons A and B with offsets wAC = 1 and wBC = 3 , respectively . Neuron C has offset value oC = 5 . Write a function to compute the output .
Codex Output	<pre>def compute_output(x1, x2, w1, w2, o1, o2, w3, w4, o3):     y1 = x1 * w1 + o1     y2 = x2 * w2 + o2     y3 = y1 * w3 + y2 * w4 + o3     return y3  print(compute_output(1, 4, 2, 1, 0.5, 1, 1, 3, 5))</pre>
Solution	22.5

Question 12 from MIT 6.036

Topic	Neural Networks II
Original Question	Neuron A takes in value negative 1 with weight 2 and offset 0.5 . Its output is passed into neuron C with weight 1 . Neuron B takes in value 1 with weight 1 and offset 1 . Its output is passed into neuron C with weight 2. Neuron C has offset 5 and a ReLU on its output . Compute the output of this neural network .
Codex Input	Neuron A takes in value -1 with weight 2 and offset 0.5 . Its output is passed into neuron C with weight 1 . Neuron B takes in value 1 with weight 1 and offset 1 . Its output is passed into neuron C with weight 2. Neuron C has offset 5 and a ReLU on its output . Write a function to compute the output of neuron C which has arguments from the output of neurons A and B. Neuron A and B do not use ReLU or max. Write functions for neurons A and B.
Codex Output	<pre>def neuronA(x):     return 2*x + 0.5  def neuronB(x):     return 1*x + 1  def neuronC(x,y):     return max(0,x+2*y+5)  print(neuronC(neuronA(-1),neuronB(1)))</pre>
Solution	7.5

Question 13 from MIT 6.036

Topic	Convolutional Neural Networks
Original Question	Given an image row [ 1 3 0 ] and filter [ 2 2 1 ], what is the result from applying the filter to the image row after applying ReLU activation on the filter's output ?
Codex Input	Given an image with values [ 1 3 0 ] and filter with weights [ 2 2 1 ], write a program to compute the result from applying the filter to the image row after applying ReLU activation on the filter's output ?
Codex Output	<pre>import numpy as np  image = np.array([[1, 3, 0]]) filter = np.array([[2, 2, 1]])  def relu(x):     return np.maximum(x, 0)  def conv(image, filter):     filter_size = filter.shape[0]     image_size = image.shape[0]     result = np.zeros((image_size - filter_size + 1))     for i in range(image_size - filter_size + 1):         result[i] = np.sum(image[i:i+filter_size] * filter)     return relu(result)  print(conv(image, filter))</pre>
Solution	8

Question 14 from MIT 6.036

Topic	State Machines and Markov Decision Processes
Original Question	Consider the input $x_t = [ 3 4 4 10 16 ]$ to a state machine with equations $s_t = f(s_{t-1}, x_t)$ and $y_t = g(s_t)$ . Compute $y_5$ if our initial conditions are $s_0$ is 7, $f(s_{t-1}, x_t) = \max(s_{t-1}, x_t)$ , and $g(s_t) = 4 * s_t$ .
Codex Input	Consider the input $x_t = [ 3 4 4 10 16 ]$ to a state machine with equations $s_t = f(s_{t-1}, x_t)$ and $y_t = g(s_t)$ . Compute $y_5$ if our initial conditions are $s_0$ is 7, $f(s_{t-1}, x_t) = \max(s_{t-1}, x_t)$ , and $g(s_t) = 4 * s_t$ .
Codex Output	<pre>def f(s_t_1, x_t):     return max(s_t_1, x_t)  def g(s_t):     return 4 * s_t  s_0 = 7 x_t = [3, 4, 4, 10, 16]  s_t = s_0 for i in range(5):     y_t = g(f(s_t, x_t[i]))     print(y_t)     s_t = f(s_t, x_t[i])</pre>
Solution	64

Question 15 from MIT 6.036

Topic	Reinforcement Learning
Original Question	After applying Q learning to $q = 6$ , what is its value? Let the $t$ be 8 and $a$ be 0.2.
Codex Input	In machine learning reinforcement learning, we use Q-learning. After applying Q-learning to $q = 6$ , what is its value? Let the $t$ be 8 and $a$ be 0.2.
Codex Output	<pre># In[ ]:  # Write your code here q = 6 t = 8 a = 0.2 q_t = q + a * (t - q) print(q_t)</pre>
Solution	6.4

Question 16 from MIT 6.036

Topic	Recurrent Neural Networks
Original Question	An RNN is defined as $s_t = w * s_{t-1} + x_t$ . If $s_0$ is 1, $w$ is 0, and $x$ is [ 0.25 0.5 ], what is $s_2$ ?
Codex Input	A recurrent neural network is defined as $s_t = w * s_{t-1} + x_t$ . If $s_0 = 1$ , $w = 0$ , $x_0 = 0.25$ , and $x_1 = 0.5$ , write a program to compute $s_2$ . Use $s_1$ as an intermediate step.
Codex Output	<pre>s_0 = 1 w = 0 x_0 = 0.25 x_1 = 0.5  s_1 = w * s_0 + x_0 s_2 = w * s_1 + x_1  print(s_2)</pre>
Solution	0.5

Question 17 from MIT 6.036

Topic	State Machines and Markov Decision Processes
Original Question	If we have a state machine, defined as $s_t = f(s_{t-1}, x_t)$ and $y_t = g(s_t)$ , where $x_t$ is the input, what is the output $y_5$ if we have $s_0$ being 11, $f(s_{t-1}, x_t) = \max(s_{t-1}, x_t)$ , $g(s_t) = 2 * s_t$ , and we input [ 4 12 5 3 16 ]?
Codex Input	If we have a state machine, defined as $s_t = f(s_{t-1}, x_t)$ and $y_t = g(s_t)$ , where $x_t$ is the input, what is the output $y_5$ if we have $s_0$ being 11, $f(s_{t-1}, x_t) = \max(s_{t-1}, x_t)$ , $g(s_t) = 2 * s_t$ , and we input [ 4 12 5 3 16 ]?
Codex Output	<pre>def f(s, x):     return max(s, x)  def g(s):     return 2 * s  s_0 = 11 x = [ 4, 12, 5, 3, 16 ]  s = s_0 for i in x:     s = f(s, i)  print(g(s))</pre>
Solution	32

Question 18 from MIT 6.036



Topic	Decision Trees
Original Question	Calculate the entropy of the left region of a 2D plane, split by a line . There are 1 points on the left side that are positive , 23 points on the right side , and 48 points total .
Codex Input	In machine learning decision trees, write a program to calculate the entropy of the left region of a 2D plane, split by a line . There is 1 point on the left side that is positive , 23 points on the right side , and 48 points total . Use p as 1/(48-23).
Codex Output	<pre>def entropy(p):     if p == 0 or p == 1:         return 0     else:         return -p*np.log2(p) - (1-p)*np.log2(1-p)  print(entropy(1/(48-23)))</pre>
Solution	0.2422921891

Question 19 from MIT 6.036

Topic	Decision Trees
Original Question	Consider a plane of 47 points , 26 of which are on the left side . Of the points on the left , 2 points are positive . Find the entropy of the left side .
Codex Input	In machine learning decision trees, consider a plane of 47 points , 26 of which are on the left side . Of the points on the left , 2 points are positive . Write a program to find the entropy of the left side . Pos is (postiiive points on the left)/(total points on the left), and neg is (total points on the left minus positive points)/(total points on the left).
Codex Output	<pre>pos = 2/26 neg = 24/26  entropy = -pos*math.log(pos,2) - neg*math.log(neg,2)  print(entropy)</pre>
Solution	0.3912435636

Question 20 from MIT 6.036

Topic	Number Theory
Original Question	Find the multiplicative inverse of 17 modulo 72 in the range $\{0, 1, \dots, 71\}$ .
Codex Input	Find the multiplicative inverse of 17 mod 72.
Correct Answer	17
Codex Output	17
Evaluation	Correct

Question 1 from 6.042 Mathematics for Computer Scientists

Topic	Number Theory
Original Question	Find a number $x \in \{0, 1, \dots, 112\}$ such that $11x \equiv 1 \pmod{113}$ .
Codex Input	Find a number $x \in \{0, 1, \dots, 112\}$ such that $11x \equiv 1 \pmod{113}$ .
Correct Answer	72
Codex Output	<pre>def find_x(n):     for x in range(n):         if (11*x) % n == 1:             return x     return None  print(find_x(113))</pre>
Evaluation	Correct

Question 2 from 6.042 Mathematics for Computer Scientists

Topic	Probability
Original Question	Finalphobia is a rare disease in which the victim has the delusion that he or she is being subjected to an intense mathematical examination. A person selected uniformly at random has finalphobia with probability 1/100. A person with finalphobia has shaky hands with probability 9/10. A person without finalphobia has shaky hands with probability 1/20. What is the probability that a person selected uniformly at random has finalphobia, given that he or she has shaky hands?
Codex Input	Finalphobia is a rare disease in which the victim has the delusion that he or she is being subjected to an intense mathematical examination. A person selected uniformly at random has finalphobia with probability 1/100. A person with finalphobia has shaky hands with probability 9/10. A person without finalphobia has shaky hands with probability 1/20. What is the probability that a person selected uniformly at random has finalphobia, given that he or she has shaky hands?
Correct Answer	18/117
Codex Output	<pre> # P(finalphobia shakyhands) = P(finalphobia and   shakyhands) / P(shakyhands) # P(finalphobia and shakyhands) = P(finalphobia) *   P(shakyhands finalphobia) # P(shakyhands) = P(shakyhands finalphobia) *   P(finalphobia) + P(shakyhands ~finalphobia) *   P(~finalphobia)  p_finalphobia = 1/100 p_shakyhands_given_finalphobia = 9/10 p_shakyhands_given_not_finalphobia = 1/20 p_not_finalphobia = 1 - p_finalphobia  p_finalphobia_and_shakyhands = p_finalphobia *   p_shakyhands_given_finalphobia p_shakyhands = p_shakyhands_given_finalphobia *   p_finalphobia + p_shakyhands_given_not_finalphobia *   p_not_finalphobia  p_finalphobia_given_shakyhands =   p_finalphobia_and_shakyhands / p_shakyhands  print(p_finalphobia_given_shakyhands) </pre>
Evaluation	Correct

Question 3 from 6.042 Mathematics for Computer Scientists

Topic	Probability
Original Question	Suppose that you roll five 6-sided dice that are fair and mutually independent. What is the probability that all five dice show different values?
Codex Input	You roll five 6-sided fair dice and mutually independent. Write a program to compute the probability that all five dice show different values.
Correct Answer	120/1296
Codex Output	<pre> import random  def roll_dice():     return random.randint(1, 6)  def roll_all_dice():     return [roll_dice() for _ in range(5)]  def all_different(dice):     return len(set(dice)) == 5  def main():     n_trials = 100000     n_success = 0     for _ in range(n_trials):         if all_different(roll_all_dice()):             n_success += 1     print(n_success / n_trials)  if __name__ == '__main__':     main() </pre>
Evaluation	Correct

Question 4 from 6.042 Mathematics for Computer Scientists

Topic	Probability
Original Question	MIT students sometimes delay laundry for a few days. Assume all random values described below are mutually independent. A busy student must complete 3 problem sets before doing laundry. Each problem set requires 1 day with probability $2/3$ and 2 days with probability $1/3$ . Let $B$ be the number of days a busy student delays laundry. What is $\text{Ex}(B)$ ?
Codex Input	MIT students sometimes delay laundry for a few days. Assume all random values described below are mutually independent. Days for illness recovery is $u=a*b$ where $a$ and $b$ are independent dice rolls. What's the expected value of $u$ ?
Correct Answer	4
Codex Output	<pre> import numpy as np  a = np.random.randint(1,7,size=100000) b = np.random.randint(1,7,size=100000)  u = a*b  print(np.mean(u)) </pre>
Evaluation	Correct

Question 5 from 6.042 Mathematics for Computer Scientists

Topic	Probability
Original Question	MIT students sometimes delay laundry for a few days. Assume all random values described below are mutually independent. A relaxed student rolls a fair, 6-sided die in the morning. If he rolls a 1, then he does his laundry immediately (with zero days of delay). Otherwise, he delays for one day and repeats the experiment the following morning. Let $R$ be the number of days a relaxed student delays laundry. What is $\text{Ex}(R)$ ?
Codex Input	Roll the 6 sided dice infinitely if you don't get a 1. Don't count the first dice roll toward days, so remember to subtract the right value from return days. return days-1. Roll infinitely until you encounter a 1. Write a program to compute the expected value for number of rolls until 1 is encountered?
Correct Answer	5
Codex Output	<pre>import random  def dice_roll():     return random.randint(1,6)  def roll_until_one():     days = 0     while True:         days += 1         if dice_roll() == 1:             return days-1  def expected_rolls():     total_days = 0     for i in range(1000000):         total_days += roll_until_one()     return total_days/1000000  print(expected_rolls())</pre>
Evaluation	Correct

Question 6 from 6.042 Mathematics for Computer Scientists

Topic	Probability
Original Question	MIT students sometimes delay laundry for a few days. Assume all random values described below are mutually independent. Before doing laundry, an unlucky student must recover from illness for a number of days equal to the product of the numbers rolled on two fair, 6-sided dice. Let $U$ be the expected number of days an unlucky student delays laundry. What is $\text{Ex}(U)$ ?
Codex Input	Roll the 2 x 6-sided dice. The product of the two dice is number of days. What is the expected value for number of days?
Correct Answer	49/4
Codex Output	<pre>import random  def roll_dice():     return random.randint(1,6), random.randint(1,6)  def expected_value(n):     total = 0     for i in range(n):         total += roll_dice()[0] * roll_dice()[1]     return total / n  print(expected_value(1000000))</pre>
Evaluation	Correct

Question 7 from 6.042 Mathematics for Computer Scientists

Topic	Probability
Original Question	I have twelve cards: 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6. I shuffle them and deal them in a row. What is the expected number of adjacent pairs with the same value?
Codex Input	I have twelve cards: 1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6. I shuffle them and deal them in a row. Write a program that computes the average number of adjacent pairs with the same value over 10000 runs
Correct Answer	1
Codex Output	<pre>import random  def run_simulation():     cards = [1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6]     random.shuffle(cards)     count = 0     for i in range(len(cards) - 1):         if cards[i] == cards[i + 1]:             count += 1     return count  def main():     total = 0     for i in range(10000):         total += run_simulation()     print(total / 10000)  if __name__ == "__main__":     main()</pre>
Evaluation	Correct

Question 8 from 6.042 Mathematics for Computer Scientists

Topic	Probability
Original Question	Scores for a final exam are given by picking an integer uniformly at random from the set 50, 51, . . . , 97, 98. The scores of all 128 students in the class are assigned in this manner. Let $S_1, \dots, S_{128}$ be their scores. For $i = 1, \dots, 128$ , what is $E[S_i]$ ?
Codex Input	Scores for a final exam are given by picking an integer uniformly at random from the set 50, 51, . . . , 97, 98. The scores of 128 students in a class are assigned in this manner. Write a program to compute the mean score?
Correct Answer	74
Codex Output	<pre>import random  def mean(n):     sum = 0     for i in range(n):         sum += random.randint(50, 98)     return sum/n  print(mean(128))</pre>
Evaluation	Correct

Question 9 from 6.042 Mathematics for Computer Scientists



Topic	Probability
Original Question	Scores for a final exam are given by picking an integer uniformly at random from the set 50, 51, . . . , 97, 98. The scores of all 128 students in the class are assigned independently in this manner. Let $S_1, \dots, S_{128}$ be their scores. Find $\text{Var}(S_i)$ .
Codex Input	Scores for a final exam are given by picking an integer uniformly at random from the set 50, 51, . . . , 97, 98. The scores of 128 students in a class are assigned independently in this manner. Write a program to compute the variance of the score.
Correct Answer	200
Codex Output	<pre>import numpy as np  def compute_variance(n):     scores = np.random.randint(50, 99, n)     return np.var(scores)  print(compute_variance(128))</pre>
Evaluation	Correct

### Question 10 from 6.042 Mathematics for Computer Scientists

Topic	Probability
Original Question	Scores for a final exam are given by picking an integer uniformly at random from the set 50, 51, . . . , 97, 98. The scores of all 128 students in the class are assigned independently in this manner. Let $S$ be the average of their scores. Find $\text{Var}(S)$ .
Codex Input	Write a function that uniformly samples 128 scores from 50,52 ... 98 and finds the mean of that sample. Then write a function that loops 100000 times and finds $\text{np.var}$ of the samples.
Correct Answer	25/16
Codex Output	<pre>def sample_mean(n):     return np.mean(np.random.choice(range(50,99), n))  def sample_var(n):     return np.var([sample_mean(n) for i in range(100000)])  print(sample_var(128))</pre>
Evaluation	Correct

### Question 11 from 6.042 Mathematics for Computer Scientists

Topic	Graph Theory
Original Question	What is the diameter of the following graph? graph = 'A': ['B'], B': ['A', 'C', 'H'], C': ['B', 'D', 'E'], D': ['C', 'E'], E': ['C', 'D', 'F', 'H'], F': ['E', 'G'], G': ['F'], H': ['B', 'E']
Codex Input	What is the diameter of the following graph? The starting node doesn't count towards length so subtract 1 from the final answer. graph = 'A': ['B'], B': ['A', 'C', 'H'], C': ['B', 'D', 'E'], D': ['C', 'E'], E': ['C', 'D', 'F', 'H'], F': ['E', 'G'], G': ['F'], H': ['B', 'E']. Remember to define the shortest path function. Use python3.
Correct Answer	5
Codex Output	<pre>def solve_problem():     """     What is the diameter of the following graph?     graph = {'A': ['B'], B': ['A', 'C', 'H'], C': ['B', 'D', 'E'],             D': ['C', 'E'], E': ['C', 'D', 'F', 'H'], F': ['E', 'G'], G': ['F'], H': ['B', 'E']}     """     graph = {'A': ['B'], 'B': ['A', 'C', 'H'], 'C': ['B', 'D', 'E'],             'D': ['C', 'E'], 'E': ['C', 'D', 'F', 'H'], 'F': ['E', 'G'], 'G': ['F'], 'H': ['B', 'E']}     return diameter(graph)  if __name__ == '__main__':     print(solve_problem())</pre>
Evaluation	Correct

### Question 12 from 6.042 Mathematics for Computer Scientists

Topic	Combinatorics
Original Question	In how many different ways can the letters in the name of the popular 1980's band BANANARAMA be arranged?
Codex Input	Write a program to compute the number of arrangements of characters in the word BANANARAMA. Remember to discount arrangements of repeated letters.
Correct Answer	$10!/(5!2!1!1!1!) = 15120$
Codex Output	<pre> def factorial(n):     if n == 0:         return 1     else:         return n * factorial(n-1)  def permutations(word):     if len(word) == 1:         return [word]     else:         result = []         for i in range(len(word)):             for p in permutations(word[:i] + word[i+1:]):                 result.append(word[i] + p)         return result  def arrangements(word):     return len(set(permutations(word)))  def main():     print(arrangements('BANANARAMA'))  if __name__ == '__main__':     main() </pre>
Evaluation	Correct

Question 13 from 6.042 Mathematics for Computer Scientists

Topic	Combinatorics
Original Question	How many different solutions over the natural numbers are there to the following equation? $x^1 + x^2 + x^3 + \dots + x^8 = 100$ A solution is a specification of the value of each variable $x_i$ . Two solutions are different if different values are specified for some variable $x_i$ .
Codex Input	Write a program to find the number of positive integer solutions to the equation $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 = 100$ . 100 is the highest number to consider for each variable.
Correct Answer	107!/100!7!
Codex Output	<pre> def main():     """     main function     """     # initialize the number of solutions     num_sol = 0     # loop through all the possible values of x1     for x1 in range(1, 101):         # loop through all the possible values of x2         for x2 in range(1, 101):             # loop through all the possible values of x3             for x3 in range(1, 101):                 # loop through all the possible values of x4                 for x4 in range(1, 101):                     # loop through all the possible values of x5                     for x5 in range(1, 101):                         # loop through all the possible values of                         # x6                         for x6 in range(1, 101):                             # loop through all the possible values                             # of x7                             for x7 in range(1, 101):                                 # loop through all the possible                                 # values of x8                                 for x8 in range(1, 101):                                     # check if the equation is                                     # satisfied                                     if x1**1+x2**2+x3**3+x4**4   +x5**5+x6**6+x7**7+x8**8 == 100:   # increment the number of   # solutions   num_sol += 1      # print the number of solutions     print(num_sol)  if __name__ == "__main__":     main() </pre>
Evaluation	Correct

Question 14 from 6.042 Mathematics for Computer Scientists

Topic	Combinatorics
Original Question	Find the coefficients of $x^{10}$ in $(x + (1/x))^{100}$
Codex Input	Using sympy, find the coefficients of $x^{10}$ in $(x + (1/x))^{100}$ . Expand and then use the result to find the coeff of $x^{10}$
Correct Answer	10055
Codex Output	<pre> from sympy import * x = symbols('x') expr = (x + 1/x)**100 print(expr) print(expand(expr)) print(expand(expr).coeff(x, 10)) </pre>
Evaluation	Correct

\*

Question 15 from 6.042 Mathematics for Computer Scientists

Topic	Propositional Logic
Original Question	Express each of the following predicates and propositions in formal logic notation. The domain of discourse is the nonnegative integers, $\mathbb{N}$ . In addition to the propositional operators, variables and quantifiers, you may define predicates using addition, multiplication, and equality symbols, but no constants (like 0, 1, . . .). For example, the proposition “n is an even number” could be written $\exists m.(m + m = n)$ . n is the sum of three perfect squares.
Codex Input	Express each of the following predicates and propositions in formal logic notation. The domain of discourse is the nonnegative integers, $\mathbb{N}$ . In addition to the propositional operators, variables and quantifiers, you may define predicates using addition, multiplication, and equality symbols, but no constants (like 0, 1, . . .). For example, the proposition “n is an even number” could be written $\exists m.(m + m = n)$ . Write a program to output 'n is the sum of three perfect squares' in formal logic.
Correct Answer	$\exists x \exists y \exists z.(x \cdot x + y \cdot y + z \cdot z = n)$
Codex Output	<pre> # n is the sum of three perfect squares # x . y . z . (x + y + z = n) # x . y . z . (x * x + y * y + z * z = n) # x . y . z . (x * x + y * y + z * z = n) </pre>
Evaluation	Correct

Question 16 from 6.042 Mathematics for Computer Scientists

Topic	Graph Theory
Original Question	Give a small example of a DAG with more than one minimum time schedule.
Codex Input	Write a program which gives a small example of a DAG with more than one minimum time schedule.
Correct Answer	Many correct answers exist.
Codex Output	<pre> # DAG = { #   'A': [('B', 2), ('C', 1)], #   'B': [('D', 2), ('E', 2)], #   'C': [('E', 1)], #   'D': [('F', 1)], #   'E': [('F', 3)], #   'F': [] # } </pre>
Evaluation	Correct

Question 17 from 6.042 Mathematics for Computer Scientists

Topic	Graph Theory
Original Question	What is the chromatic number of the following graph? graph = 'A': ['B'], B': ['A', 'C', 'H'], C': ['B', 'D', 'E'], D': ['C', 'E'], E': ['C', 'D', 'F', 'H'], F': ['E', 'G'], G': ['F'], H': ['B', 'E']
Codex Input	Find the chromatic number of a graph with adjacency matrix: [[0, 1, 0, 0, 0, 0, 0, 0], [1, 0, 1, 0, 0, 0, 0, 1], [0, 1, 0, 1, 1, 0, 0, 0], [0, 0, 1, 0, 1, 0, 0, 0], [0, 0, 1, 1, 0, 1, 0, 1], [0, 0, 0, 0, 1, 0, 1, 0], [0, 0, 0, 0, 0, 1, 0, 0], [0, 1, 0, 0, 1, 0, 0, 0]]""Find the chromatic number of a graph with adjacency matrix: [[0, 1, 0, 0, 0, 0, 0, 0], [1, 0, 1, 0, 0, 0, 0, 1], [0, 1, 0, 1, 1, 0, 0, 0], [0, 0, 1, 0, 1, 0, 0, 0], [0, 0, 1, 1, 0, 1, 0, 1], [0, 0, 0, 0, 1, 0, 1, 0], [0, 0, 0, 0, 0, 1, 0, 0], [0, 1, 0, 0, 1, 0, 0, 0]]
Correct Answer	3
Codex Output	<pre> from networkx import from_numpy_matrix from networkx.algorithms.coloring import greedy_color from numpy import array  A = array([[0, 1, 0, 0, 0, 0, 0, 0],            [1, 0, 1, 0, 0, 0, 0, 1],            [0, 1, 0, 1, 1, 0, 0, 0],            [0, 0, 1, 0, 1, 0, 0, 0],            [0, 0, 1, 1, 0, 1, 0, 1],            [0, 0, 0, 0, 1, 0, 1, 0],            [0, 0, 0, 0, 0, 1, 0, 0],            [0, 1, 0, 0, 1, 0, 0, 0]])  G = from_numpy_matrix(A)  print(greedy_color(G, strategy='largest_first')) </pre>
Evaluation	Correct

Question 18 from 6.042 Mathematics for Computer Scientists

Topic	Set Theory
Original Question	Define a bijection between the natural numbers and the integers.
Codex Input	To define a bijection between natural number and integers, map the even and odd natural numbers to positive and negative integers. Divide by 2 and rounding down.
Correct Answer	There are uncountably many correct answers
Codex Output	<pre> def even_odd_to_int(n):     if n % 2 == 0:         return n // 2     else:         return -(n + 1) // 2 </pre>
Evaluation	Correct

Question 19 from 6.042 Mathematics for Computer Scientists

Topic	Combinatorics
Original Question	How many different paths are there from point (0, 0, 0) to point (12, 24, 36) if every step increments one coordinate and leaves the other two unchanged?
Codex Input	How many different paths are there from point (0, 0, 0) to point (12, 24, 36) if every step increments one coordinate and leaves the other two unchanged?
Correct Answer	72!/12!24!36!
Codex Output	<pre>def solve_problem():     """     How many different paths are there from point (0, 0, 0)     to point (12, 24, 36) if every step increments one     coordinate and leaves the other two unchanged?     """     return factorial(12 + 24 + 36) / (factorial(12) *         factorial(24) * factorial(36))  if __name__ == '__main__':     print(solve_problem())</pre>
Evaluation	Correct

Question 20 from 6.042 Mathematics for Computer Scientists

Topic	Cosmology
Original Question	Derive the ratio between the stellar mass density $\rho_*$ and the current critical density. Assume that the density of galaxies is $2 \times 10^{-2} \text{ Mpc}^{-3}$ , that in each galaxy there are $5 \times 10^{10}$ stars, and that each star has an average mass of 0.5 M. Assume that the current critical density, $\rho_{c,0}$ , is $1.4 \times 10^{11} \frac{M}{\text{Mpc}^3}$ .
Codex Input	Write a program that calculates the ratio between the stellar mass density $\rho_*$ and the current critical density. Assume that the density of galaxies is $2 \times 10^{-2} \text{ Mpc}^{-3}$ , that in each galaxy there are $5 \times 10^{10}$ stars, and that each star has an average mass of 0.5 M. Assume that the current critical density, $\rho_{c,0}$ , is $1.4 \times 10^{11} \frac{M}{\text{Mpc}^3}$ . $\rho_*$ is the product of the density times the stars in each galaxy times the mass of each star.
Correct Answer	$3.6 \times 10^{-3}$
Codex Output	<pre> # Define the density of galaxies rho_galaxies = 2e-2  # Define the number of stars in each galaxy n_stars = 5e10  # Define the mass of each star m_star = 0.5  # Define the current critical density rho_crit = 1.4e11  # Calculate the stellar mass density rho_star = rho_galaxies * n_stars * m_star  # Calculate the ratio between the stellar mass density and the current critical density rho_ratio = rho_star / rho_crit  # Print the result print(rho_ratio) </pre>
Evaluation	Correct

Question 1 from 8.282 Introduction to Astronomy



Topic	angular resolution
Original Question	Calculate the best angular resolution that can, in principle, be achieved with the human eye. Assume a pupil diameter of 0.5 cm and the wavelength of green light, $\sim 0.5\mu\text{m}$ . Express your answer in arcminutes, where an arcminute is $1/60$ of a degree.
Codex Input	Write a program to calculate the best angular resolution that can be achieved with the human eye. Assume a pupil of circular aperture (use constant) has diameter of 0.5 centimeter and the light has wavelength 0.5 micrometer. Express your answer in arcminutes, where an arcminute is $1/60$ of a degree. Radians to degrees is 57.2958. Degree to arcminute is 60. Convert micrometer and centimeter to meter.
Correct Answer	25 m arcsec or 0.42 arcmins
Codex Output	<pre> import math  # constants pupil_diameter = 0.5 # cm light_wavelength = 0.5 # micrometer  # convert to meter pupil_diameter = pupil_diameter / 100 # cm to m light_wavelength = light_wavelength / 1e6 # micrometer to m  # calculate angular resolution angular_resolution = 1.22 * light_wavelength /     pupil_diameter  # convert to arcminute angular_resolution = angular_resolution * 57.2958 * 60 #     rad to arcminute  print(angular_resolution) </pre>
Evaluation	Correct

Question 2 from 8.282 Introduction to Astronomy

Topic	angular resolution
Original Question	A triple stellar system, showing a parallax of 0.01 arcsecond, is composed of a binary system, with a distance between the two stars of 0.2 AU, and a third star, at a distance of 90 AU from the binary. This system is observed in the optical ( $\lambda \sim 5500 \text{ \AA}$ ) with a 1 meter telescope. What is the distance of this system ?
Codex Input	A triple stellar system, showing a parallax of 0.01 arcsecond, is composed of a binary system, with a distance between the two stars of 0.2 AU, and a third star, at a distance of 90 AU from the binary. Write a program that computes the distance of this system by taking the inverse of the parallax.
Correct Answer	100 pc
Codex Output	<pre> # Solution # # The distance to the system can be calculated by taking #   the inverse of the parallax. # # \$\$d = \frac{1}{\varpi}\$\$ # # where <math>\varpi</math> is the parallax in arcseconds. # # The parallax of the system is 0.01 arcseconds, so the #   distance is: # # \$\$d = \frac{1}{0.01}\$\$ # # \$\$d = 100\$\$ # # \$\$d = 100 \text{ AU}\$\$ # # The distance to the system is 100 AU. </pre>
Evaluation	Correct

Question 3 from 8.282 Introduction to Astronomy

Topic	general
Original Question	What is the geometrical cross section of the Earth given its radius $R$ earth?
Codex Input	Write a program that computes the area of a circle whose radius is the radius of the Earth in $\text{cm}^2$ .
Correct Answer	$1.3 \times 10^{18}$
Codex Output	<pre> "# Earth's radius in cm radius = 6.37 * 10**8  # Compute the area of the circle area = 3.14 * radius**2  # Print the area print(area) " </pre>
Evaluation	Correct

Question 4 from 8.282 Introduction to Astronomy

Topic	general
Original Question	In this question we will find what is the estimated rate for star to star collision. For simplicity we assume that all the stars in our galaxy have the same properties as our Sun, i.e., the same solar radius, and the same velocity, of $200 \text{ km sec}^{-1}$ , and that the direction of their velocity is random (this of-course is <i>not</i> true, but sufficient for this rough estimation). Assume the stars number density in the galaxy is $10^{-9} \text{ pc}^{-3}$ and neglect the gravitational pull between the stars. Find the mean free path between collision to collision.
Codex Input	Assume the stars number density in the galaxy is $10^{-9}$ . Write a program that computes the mean free path between collision to collision in parsecs. Pi is 3.1416. Radius of the Sun is $6.9551 * 10^{10} \text{ cm}$ . Divide (parsec in centimeters) <sup>2</sup> by $\pi * r^2 * n$ . Parsec is $3.0857 * 10^{18} \text{ cm}$ .
Correct Answer	$6.27 \times 10^{23} \text{ rmpc}$
Codex Output	<pre>import math  n = 10 ** (-9) r = 6.9551 * 10 ** 10  mfp = (3.0857 * 10 ** 18) ** 2 / (math.pi * r ** 2 * n)  print(mfp)</pre>
Evaluation	Correct

Question 5 from 8.282 Introduction to Astronomy

Topic	Hubble's Law
Original Question	Measurements of the radial recession velocity of five galaxies in a cluster give velocities of 9700, 8600, 8200, 8500 and 10000 $\text{km s}^{-1}$ . What is the distance to the cluster if the Hubble parameter is $H_0 = 72 \text{ km s}^{-1} \text{ Mpc}^{-1}$ ? Hint: Use the Hubble law for the average velocity of the members in the cluster.
Codex Input	Measurements of the velocity of five galaxies are 9700, 8600, 8200, 8500 and 10000 $\text{km s}^{-1}$ . Write a program to find the distance to the cluster if the Hubble parameter is $H_0 = 72 \text{ km s}^{-1} \text{ Mpc}^{-1}$ ? Divide the average velocity by the Hubble parameter.
Correct Answer	125 Mpc
Codex Output	<pre># Solution:  # Given data  H_0 = 72 # Hubble parameter v = [9700, 8600, 8200, 8500, 10000] # velocities of     galaxies in km/s  # Calculations  v_avg = sum(v)/len(v) # average velocity d = v_avg/H_0 # distance in Mpc  # Results  print("""The distance to the cluster is""", d, ""Mpc""")  ''' The Hubble parameter is given by H_0 = v_avg/d '''</pre>
Evaluation	Correct

Question 6 from 8.282 Introduction to Astronomy

Topic	Neutron Stars
Original Question	A neutron star whose mass is $M = 1.4M_{\odot}$ and radius is $R = 10$ km is in orbit about a normal star. Matter is flowing from the normal star onto the collapsed star at a rate of $\dot{M} = 10^{17}$ grams per second. Assume that all of the gravitational potential energy of the infalling matter is converted to radiation. Compute the bolometric luminosity of the neutron star.
Codex Input	A neutron star has mass $M = 1.4$ times $1.989e33$ , and radius is $R = 1000000$ cm. Matter is flowing from a star the neutron star at a rate of $\dot{M} = 10^{17}$ grams per second. Write a function to compute the bolometric luminosity of the neutron star. Gravitational constant is $6.673 * 10^{-8}$ dyne cm <sup>2</sup> g <sup>-2</sup> . It should return $G$ times $M$ times $\dot{M}$ divided by $R$ .
Correct Answer	$1.9 \times 10^{37}$ ergs/sec
Codex Output	<pre>def luminosity(M, R, dotM):     G = 6.673e-8     return G * M * dotM / R  print(luminosity(1.4 * 1.989e33, 1000000, 1e17))</pre>
Evaluation	Correct

Question 7 from 8.282 Introduction to Astronomy

Topic	Neutron Stars
Original Question	A neutron star whose mass is $1.4M_{\odot}$ is in orbit about a normal star. Matter is flowing from the normal star onto the collapsed star at a rate of $\dot{M} = 10^{17}$ grams per second. Assume that all of the gravitational potential energy of the infalling matter is converted to radiation. Suppose that the surface of the neutron star radiates this energy as a blackbody of a single temperature $T$ . Compute $T$ (Assume the radius of the neutron star, $R$ , is 10 km).
Codex Input	A neutron star has mass $M = 1.4$ times $1.989e33$ , and radius is $R = 1000000$ cm. Matter is flowing from a star the neutron star at a rate of $\dot{M} = 10^{17}$ grams per second. Write a function to compute the bolometric luminosity of the neutron star. Gravitational constant is $6.673 * 10^{-8}$ dyne cm <sup>2</sup> g <sup>-2</sup> . It should return $G$ times $M$ times $\dot{M}$ divided by $R$ . Then, use this value to calculate the radiated blackbody temperature.
Correct Answer	$12.7 \times 10^6$ K
Codex Output	<pre>def Lbol(M, R, dotM):     G = 6.673e-8     return G * M * dotM / R  def Tbb(L, R):     sb = 5.67e-5     return (L / (4 * np.pi * R**2 * sb))**0.25  print(Tbb(Lbol(1.4 * 1.989e33, 1000000, 1e17), 1000000))</pre>
Evaluation	Correct

Question 8 from 8.282 Introduction to Astronomy

Topic	Neutron Stars
Original Question	An X-ray binary has an orbital period of $P = 42$ minutes. The neutron star moves in a circular orbit about the center of mass of the binary with a velocity of $V_x = 11\text{km/s}$ , while its unseen companion star has an orbital velocity of $V_c = 770\text{km/s}$ . Find the masses of the unseen companion star and the neutron star.
Codex Input	An X-ray binary has an orbital period of $P=42*60$ seconds. The neutron star moves in a circular orbit about the center of mass of the binary with a velocity of $V_x = 1100000$ cm/s, while its unseen companion star has an orbital velocity of $V_c = 77000000$ cm/s. Write a function to find the masses of the unseen companion star and the neutron star. Pi is 3.1416. Gravitational constant is $6.673 * 10^{-8}$ dyne $\text{cm}^2 \text{g}^{-2}$ . Compute $((4*\pi^2)/(P^2) * (a^3)/(G)) / (1 + ((v_c * P / (2 * \pi)) / (v_x * P / (2 * \pi)))) / (\text{mass of the Sun})$ for the mass of the companion star. Let the mass of the Sun be $1.989 * 10^{33}$ .
Correct Answer	$M_c = 0.02M_\odot$ ; $M_x = 1.415M_\odot$
Codex Output	<pre>def mass_of_star(P, v_x, v_c):     G = 6.673 * 10 ** -8     M_sun = 1.989 * 10 ** 33     return ((4 * np.pi ** 2) / (P ** 2) * (a ** 3) / (G)) /             (1 + ((v_c * P / (2 * np.pi)) / (v_x * P / (2 *             np.pi)))) / (M_sun)  print(mass_of_star(42*60, 1100000, 77000000))</pre>
Evaluation	Correct

Question 9 from 8.282 Introduction to Astronomy

Topic	Neutron stars
Original Question	Evaluate the minimum rotation period of a neutron star, meaning the fastest it can spin without breaking apart. Assume a mass of $1.4 M_\odot$ , a 10 km radius, and that the star remains spherical.
Codex Input	Evaluate the minimum rotation period of a neutron star, meaning the fastest it can spin without breaking apart. Assume a mass of 1.4 times the mass of the sun ( $1.989 * 10^{33}$ ), a 10 km radius, and that the star remains spherical. Pi is 3.1416. Gravitational constant is $6.673 * 10^{-8}$ dyne $\text{cm}^2 \text{g}^{-2}$ . Calculate this by using $(R^3 * (2*\pi)^2 / (G*M))^{1/2}$ .
Correct Answer	$P > 0.5\text{ms}$
Codex Output	<pre># Define the variables  G = 6.673 * 10 ** -8 M = 1.989 * 10 ** 33 R = 10 * 10 ** 5  # Calculate the minimum period  P = (R ** 3 * (2 * np.pi) ** 2 / (G * M)) ** (1 / 2)  # Print the result  print(P)</pre>
Evaluation	Correct

Question 10 from 8.282 Introduction to Astronomy

Topic	General
Original Question	A globular cluster contains $N = 10^6$ stars. Assume they all have the mass of the Sun, $m = M_{Sun}$ , and the same speed of $v = 10\text{km/s}$ . Use the Virial Theorem to estimate a characteristic size of the cluster.
Codex Input	A globular cluster contains $N = 10^6$ stars. Assume they all have the mass of the Sun, $m = M_{Sun}$ , and the same speed of $v = 10\text{ km/s}$ . Write a function that uses the Virial Theorem ( $GNm / v^2$ ) to estimate a characteristic size of the cluster and divides the answer by $3.08567758e18$ . Gravitational constant is $6.673 * 10^{-8}\text{ dyne cm}^2\text{ g}^{-2}$ . Mass of the Sun is $1.9891 * 10^{33}\text{ g}$ .
Correct Answer	$R = 40pc$
Codex Output	<pre>def virial_theorem(N, m, v):     return (6.673e-8 * N * m / v**2) / 3.08567758e18  print(virial_theorem(1e6, 1.9891e33, 10e5))</pre>
Evaluation	Correct

Question 11 from 8.282 Introduction to Astronomy

Topic	General
Original Question	A reasonable approximation to the $M_V - P$ (absolute brightness - period) relation for Cepheid variable is: $M_V = -2.3\log_{10}(P/\text{days}) - 1.7$ . A Cepheid variable with mean apparent magnitude of $V = 19.0$ is found in the galaxy M31 with a pulsation period of 30 days. Find the distance to M31 in parsecs.
Codex Input	A reasonable approximation to the $M_V - P$ (absolute brightness - period) relation for Cepheid variable is: $M_V = -2.3\log_{10}(P/\text{days}) - 1.7$ . A Cepheid variable with mean apparent magnitude of $V=19.0$ is found in the galaxy M31 with a pulsation period of 30 days. Write a function that solves $10^{((V - M_V)/5 + 1)}$ .
Correct Answer	$d = 660,700pc$
Codex Output	<pre>def period(V, M_V):     return 10**((V - M_V)/5 + 1)  print(period(19.0, -2.3*np.log10(30) - 1.7))</pre>
Evaluation	Correct

Question 12 from 8.282 Introduction to Astronomy

Topic	General
Original Question	A certain red giant star has a radius that is 500 times that of the Sun, and a temperature that is 1/2 that of the Sun's temperature. Find the bolometric (total) luminosity in units of the bolometric luminosity of the Sun.
Codex Input	A certain red giant star has a radius that is 500 times that of the Sun, and a temperature that is 1/2 that of the Sun's temperature of 5778 K. Find the bolometric (total) luminosity in units of the bolometric luminosity of the Sun.
Correct Answer	$L_{bol,redgiant} = 1.56x10^4 L_{bol,sun}$
Codex Output	<pre># Importing Modules import numpy as np  # Declaring variables R = 500 * 6.957e8 T = 0.5 * 5778  # Calculations L = 4 * np.pi * R**2 * 5.67e-8 * T**4  # Results print('The bolometric luminosity is', L/3.828e26, 'times       that of the Sun.')</pre>
Evaluation	Correct

Question 13 from 8.282 Introduction to Astronomy

Topic	General
Original Question	It has long been suspected that there is a massive black hole near the center of our Galaxy. Recently, a group of astronomers determined the parameters of a star that is orbiting the suspected black hole. The orbital period is $P = 15$ years, and the orbit has an angular radius of $\theta = 0.12$ arcsec. Take the distance to the Galactic center to be $D_{galacticcenter} = 8\text{kpc}$ . Compute the mass of the black hole, starting from $F=ma$ . Express the answer in units of the Sun's mass. (assume that Newton's law of gravity is applicable for orbits sufficiently far from a black hole, and that the orbiting star satisfies this condition.
Codex Input	Find the gravitational acceleration of Earth due to the Sun. Gravitational constant is $6.673 * 10^{-8}$ dyne $\text{cm}^2 \text{g}^{-2}$ . Mass of the Sun is $1.9891 * 10^{33}$ g.
Correct Answer	$M_{BH} = 3.9 * 10^6 M_{Sun}$
Codex Output	<pre>G = 6.673 * 10 ** -8 M = 1.9891 * 10 ** 33 r = 1.5 * 10 ** 13  a = G * M / r ** 2  print(a)</pre>
Evaluation	Correct

Question 14 from 8.282 Introduction to Astronomy

Topic	General
Original Question	A galaxy moves directly away from us with a speed of $v = 3000$ km/s. Find the observed wavelength of the $H_{\alpha}$ line in that galaxy's spectrum. The rest wavelength of $H_{\alpha}$ is $\lambda_{rest} = 6565$ Angstrom.
Codex Input	A galaxy moves directly away from us with a speed of $v = 3000$ km/s. Find the observed wavelength of the $H_{\alpha}$ line in that galaxy's spectrum. The rest wavelength of $H_{\alpha}$ is $\lambda_{rest} = 6565$ Angstrom.
Correct Answer	$\lambda_{observed} = 6630$ Angstrom
Codex Output	<pre># Import modules import numpy as np  # Define constants c = 3e5 # km/s  # Define the function def wavelength(v, l_rest):     return l_rest * (1 + v/c)  # Calculate the wavelength l_obs = wavelength(3000, 6565)  # Print the results print('The observed wavelength of the H_alpha line in that       galaxy\'s spectrum is {:.2f} Angstrom.'.format(l_obs))</pre>
Evaluation	Correct

Question 15 from 8.282 Introduction to Astronomy

Topic	Blackbody
Original Question	A white dwarf star has an effective temperature, $T = 50,000$ degrees Kelvin, but its radius, $R_{WD}$ , is comparable to that of the Earth. Take $R_{WD} = 10^4$ km. Compute the luminosity (power output) of the white dwarf. Treat the white dwarf as a blackbody radiator.
Codex Input	A white dwarf star has an effective temperature, $T = 50,000$ degrees Kelvin, but its radius, $R_{WD}$ , is comparable to that of the Earth. Take $R_{WD} = 10^9$ cm. Compute the luminosity (power output) of the white dwarf. Treat the white dwarf as a blackbody radiator. Pi is 3.1416. Stefan-Boltzmann constant is $5.670 * 10^{-5} \text{ g s}^{-3}$ .
Correct Answer	$L = 4.5 * 10^{33} \text{ erg / s}$
Codex Output	<pre> # Input T = 50000 R_WD = 10**9  # Function L = 4 * np.pi * R_WD**2 * 5.670 * 10**-5 * T**4  # Output print(L) </pre>
Evaluation	Correct

Question 16 from 8.282 Introduction to Astronomy



Topic	General
Original Question	Find the gravitational acceleration of Earth due to the Sun.
Codex Input	Find the gravitational acceleration of Earth due to the Sun.
Correct Answer	$a = 0.6\text{cm/s}^2$
Codex Output	<pre> import numpy as np import astropy.units as u from astropy.constants import G  # The Sun's mass in kg Msun = 1.989e30  # The Earth's distance from the Sun in m AU = 1.496e11  def gravitational_acceleration(M, r):     """     Calculate the gravitational acceleration of an object     of mass M at a distance r from the object.      Parameters     -----     M : float         Mass of the gravitating body in kg.     r : float         Distance of the object from the body in m.      Returns     -----     a : astropy.units.Quantity         The gravitational acceleration in m/s^2.     """     return G * M / r**2  def main():     """     Calculate the gravitational acceleration of the Earth     due to the Sun.     """     # Calculate the acceleration due to the Sun     a = gravitational_acceleration(Msun, AU)      # Print the result     print("The gravitational acceleration of the Earth due     to the Sun is "           "{0:.3g}".format(a))  if __name__ == '__main__':     main() </pre>
Evaluation	Correct

Question 17 from 8.282 Introduction to Astronomy

Topic	Blackbody
Original Question	A very hot star is detected in the galaxy M31 located at a distance of $d = 800$ kpc. The star has a temperature $T = 6 * 10^5$ K and a produces a flux of $f = 10^{-12} \text{ergs/cm}^2/\text{sec}$ at the Earth. Treat the star's surface as a blackbody radiator. Compute the star's radius in units of the Sun's radius.
Codex Input	A very hot star is detected in the galaxy M31 located at a distance of $d=800000$ times $3.08567758e16$ . The star has a temperature $T = 6 * 10^5$ K and a produces a flux of $f = 1^{(-11)}$ ergs / $\text{cm}^2$ / sec at the Earth. Treat the star's surface as a blackbody radiator. Compute the star's radius in units of the Sun's radius. Pi is 3.1416. Stefan-Boltzmann constant is $5.670 * 10^{-8} \text{g s}^{-3}$ . Take the square root of $((4*\pi*d^2*f)$ divided by $(4*\pi*\sigma*T^4)$ ). Finally, divide the answer by (the radius of the Sun * 100 (in centimeters)).
Correct Answer	$R = 0.012R_{Sun}$
Codex Output	<pre>import math  # Constants sigma = 5.670 * 10 ** -8  # Inputs d = 800000 * 3.08567758e16 T = 6 * 10 ** 5 f = 1 * 10 ** -11  # Calculations R = math.sqrt((4 * math.pi * d ** 2 * f) / (4 * math.pi *     sigma * T ** 4)) / (100 * 6.955e8)  # Outputs print(R)</pre>
Evaluation	Correct

Question 18 from 8.282 Introduction to Astronomy

Topic	General
Original Question	If the Sun (more precisely, the local standard of rest) is orbiting the Galactic center at $v = 220$ km/sec, estimate the mass of the Galaxy interior to the Sun's orbit (in units of $M_{Sun}$ ). Assume the Sun's distance to the center of the galaxy is $R = 8$ kpc.
Codex Input	If the Sun (more precisely, the local standard of rest) is orbiting the Galactic center at $v=22000000$ cm/sec, write a function to calculate the mass of the Galaxy interior to the Sun's orbit (in units of $M_{Sun}$ ). Assume the Sun's distance to the center of the galaxy is $R = 8000$ times $3.08567758e18$ . Gravitational constant is $6.673 * 10^{-8} \text{dyne cm}^2 \text{g}^{-2}$ .
Correct Answer	$M(< R) = 9 * 10^{10} M_{Sun}$
Codex Output	<pre>def mass_galaxy(v, r):     G = 6.673e-8     M_sun = 1.989e33     return v**2 * r / G / M_sun  print(mass_galaxy(22000000, 8000 * 3.08567758e18))</pre>
Evaluation	Correct

Question 19 from 8.282 Introduction to Astronomy

Topic	General
Original Question	If the Sun (more precisely, the local standard of rest) is orbiting the Galactic center at $v = 220$ km/sec, estimate the mass of the Galaxy interior to the Sun's orbit (in units of $M_{Sun}$ ). Assume the Sun's distance to the center of the galaxy is $R = 8$ kpc.
Codex Input	If the Sun (more precisely, the local standard of rest) is orbiting the Galactic center at $v=22000000$ cm/sec, write a function to calculate the mass of the Galaxy interior to the Sun's orbit (in units of $M_{Sun}$ ). Assume the Sun's distance to the center of the galaxy is $R = 8000$ times $3.08567758e18$ . Gravitational constant is $6.673 * 10^{-8}$ dyne cm <sup>2</sup> g <sup>-2</sup> .
Correct Answer	$M(< R) = 9 * 10^{10} M_{Sun}$
Codex Output	<pre>def mass_galaxy(v, r):     G = 6.673e-8     M_sun = 1.989e33     return v**2 * r / G / M_sun  print(mass_galaxy(22000000, 8000 * 3.08567758e18))</pre>
Evaluation	Correct

Question 20 from 8.282 Introduction to Astronomy

Topic	Market Structures
Original Question	Consider the perfectly competitive market for gasoline. The aggregate demand for gasoline is $D(p) = 100 - p$ while the aggregate supply is $S(p) = 3 * p$ . Calculate the equilibrium price and quantity. At this equilibrium, compute the consumer surplus, producer surplus and total surplus.
Codex Input	In Principles of Microeconomics, given the aggregate demand for gasoline is $D(p) = 100 - p$ and the aggregate supply is $S(k) = 3 * k$ , where $p, k > 0$ . Write a program to compute the equilibrium price and quantity. Using the equilibrium price, equilibrium quantity, and the demand at $p=0$ , write a program to compute the consumer surplus.
Codex Output	<pre>def demand(p):     return 100 - p def supply(p):     return 3 * p def equilibrium(p):     return demand(p) == supply(p) def sup_curve(p):     return supply(p) def dem_curve(p):     return demand(p) def price():     p = 0     while equilibrium(p) is False:         p = p + 1     return p def quantity():     p = price()     return supply(p) # Print the equilibrium price print(f"Equilibrium price: {price()}") # Print the equilibrium quantity print(f"Equilibrium quantity: {quantity()}")  def demand(p):     return 100 - p def supply(p):     return 3 * p def equilibrium(p):     return demand(p) == supply(p) def sup_curve(p):     return supply(p) def dem_curve(p):     return demand(p) def price():     p = 0     while equilibrium(p) is False:         p = p + 1     return p def quantity():     p = price()     return supply(p) def surplus(p):     q = quantity()     return (100 - p) * q / 2 # Print the consumer surplus print(f"Consumer surplus: {surplus(price())}")</pre>
Solution	Equilibrium Price = 25, Equilibrium Quantity = 75, Consumer Surplus = 2812.5, Producer Surplus = 937.5, Total Surplus = 3750

Question 1 from MIT 14.01

Topic	Market Structures
Original Question	A monopoly faces market demand $Q = 30 - P$ and has a cost function $C(Q) = 1/2 * (Q^2)$ . Find the profit maximizing price and quantity and the resulting profit to the monopoly.
Codex Input	In Microeconomics, a monopoly faces market demand $Q = 30 - P$ and has a cost function $C(Q) = 1/2 * (Q^2)$ . Write a program using scipy to compute the profit maximizing price and quantity to the monopoly.
Codex Output	<pre>import numpy as np from scipy import optimize  def demand(p, Q):     return 30 - p  def cost(q, Q):     return 1/2*(q**2)  def profit(p, Q):     return demand(p, Q)*p - cost(demand(p, Q), Q)  def main():     Q = 30     p = optimize.fminbound(lambda p: -profit(p, Q), 0, 30)     print(f"Optimal price is {p}")     print(f"Optimal quantity is {demand(p, Q)}")     print(f"Optimal profit is {profit(p, Q)}")  if __name__ == "__main__":     main()</pre>
Solution	Profit Maximizing Price = 20, Quantity = 10, Profit = 150

Question 2 from MIT 14.01

Topic	Market Structures
Original Question	Oliver has an endowment of \$10,000 that he wants to invest. He can either invest in a bond, which yields 1% or the stock market, which consist of one firm, Amazon. Amazon's stock costs \$100 today, and will be worth \$400 in one year with probability 0.5 or will drop to \$0 with probability 0.5. Oliver's utility function is $U(w) = \sqrt{w}$ . Due to institutional regulations, Oliver can invest only in bonds, or only in Amazon, he cannot buy both Amazon stock and bonds. What is Oliver's utility of buying bonds? What if he invests only in Amazon's stock? What does he prefer?
Codex Input	In Microeconomics, Oliver has an endowment of \$10,000 that he wants to invest. Oliver can invest in a bond with a 1% return or in Amazon stock which costs \$100 today and will be \$400 in one year with probability 0.5 or be \$0 with probability 0.5. Oliver's utility function is $U(w) = \sqrt{w}$ . Write a program to compute Oliver's utility of buying a bond.
Codex Output	<pre> import numpy as np from scipy.stats import norm  def utility(w):     return np.sqrt(w)  def bond_utility(x):     return utility(x) - 0.01*x # x is the amount of money     invested in the bond, which has a 1% return rate.  def stock_utility(x): # x is the amount of money invested     in Amazon stock, which will be \$400 in one year with     probability 0.5 or be \$0 with probability 0.5.     if x == 400: # if the stock price is \$400, then Oliver's         utility will be sqrt(400) + 0 = sqrt(400) =         10.65685424949238 (the same as utility function).         return 10.65685424949238 # If the stock price is \$0,         then Oliver's utility will be 0.     else:         return utility(x) # if the stock price is not \$400,         then Oliver's utility will be sqrt(0) = 0.  def expected_utility(x): # x is the amount of money     invested in the bond or Amazon stock.     return 0.5*stock_utility(400*norm.cdf(0.5)) +     0.5*stock_utility(0) + 0.01*x # The expected utility     of investing in a bond or Amazon stock is the sum of     the expected utility of investing in a bond and the     expected utility of investing in Amazon stock,     weighted by their probabilities of occurring (0.01).  print(bond_utility(10000)) # The expected utility of     investing in a bond is -0.01*10000 = -1. print(stock_utility(10000)) # The expected utility of     investing in Amazon stock is 0. print(expected_utility(10000)) # The expected utility of     investing in a bond or Amazon stock is the sum of the     expected utility of investing in a bond and the     expected utility of investing in Amazon stock, weighted     by their probabilities of occurring (0.01). Therefore,     the expected utility of investing in a bond or Amazon     stock is -1 + 0 = -1. </pre>
Solution	Utility of buying bonds = 100.5, Utility of investing = 100, Prefer to buy bonds.

Question 3 from MIT 14.01

Topic	Market Structures
Original Question	A uniform pricing monopolist has a cost function $C(q) = (1/2) * q^2$ . It faces a market demand of $D(p) = p^{-u}$ where $u > 1$ . Calculate the price elasticity of demand.
Codex Input	Market demand is $D(p) = p^{*(-u)}$ . Use Sympy, find an expression for the elasticity of demand using D(p), p, and u and then normalize by D(p) and multiply by p to find the percentage elasticity of demand.
Codex Output	<pre> # Solution  # Import sympy module import sympy as sym  # Define symbols p, u = sym.symbols('p u')  # Define demand function D = p**(-u)  # Find elasticity of demand elasticity = sym.diff(D, p)  # Normalize by demand elasticity = elasticity / D  # Multiply by price elasticity = elasticity * p  # Print result print(elasticity) </pre>
Solution	Optimal Price = -u

Question 4 from MIT 14.01

Topic	Pricing
Original Question	Suppose there are only two goods in the world: tea and coffee. In both the US one pound of tea requires 3 hours of labor to produce and one pound of coffee requires 2 hours of labor to produce. A worker can choose to work either in the tea industry or in the coffee industry (skills are completely transferable across industries) and consider the case when the labor market is perfectly competitive, and the market for tea and coffee are also perfectly competitive. Suppose that on the international market, due to the different production functions by different countries, we can trade k pounds of tea for 1 pound of coffee. For what values of k will the US choose to export tea? For what values of k will the US choose to export coffee
Codex Input	Set ratio 1 tea for 3 labor. Set ratio 1 coffee for 2 labor. Set international trade ratio k tea for 1 coffee. Use Sympy to find the values of k where the US will switch from importing tea to exporting.
Codex Output	<pre> from sympy import *  # Define symbols t, c, k = symbols('t c k')  # Define equations tea = t/3 coffee = c/2  # Define equation to solve eq = Eq(tea, k*coffee)  # Solve equation solve(eq, k)  # Answer: k = 2/3 </pre>
Solution	The US will export coffee for $k > 2/3$ and the US will export tea for $k < 2/3$ .

Question 5 from MIT 14.01

Topic	Production and Costs
Original Question	Is the production function: $F(L, K) = L^2 * K^{1/2}$ exhibiting constant, increasing or decreasing returns to scale.
Codex Input	In Principles of Microeconomics, Write a program to check if the production function $F(L, K) = L^2 * K^{1/2}$ is exhibiting constant returns to scale. In Principles of Microeconomics, Write a program to check if the production function $F(L, K) = L^2 * K^{1/2}$ has increasing returns to scale.
Codex Output	<pre> # Import packages import numpy as np  # Define production function def production_function(L, K):     return L**2 * K**(1/2)  # Define parameters L = np.array([1, 2, 3, 4, 5]) K = np.array([1, 2, 3, 4, 5])  # Check if production function exhibits constant returns to scale if np.all(production_function(L, K) == production_function(2*L, 2*K)):     print("Production function exhibits constant returns to scale.") else:     print("Production function does not exhibit constant returns to scale.")  def F(L, K):     return L**2 * K**(1/2)  def check_increasing_returns(L, K):     if F(L+1, K) &gt; F(L, K):         return True     else:         return False  print(check_increasing_returns(1, 1)) print(check_increasing_returns(1, 2)) print(check_increasing_returns(2, 1)) print(check_increasing_returns(2, 2)) </pre>
Solution	Increasing Returns to Scale

Question 6 from MIT 14.01



Topic	Production and Costs
Original Question	For each of the following production functions: (a) $F(L, K) = L^2 * K^{1/2}$ (b) $F(L, K) = L + L^{1/2} * K^{1/2}$ (c) $F(L, K) = 2 * L + K$ State whether the production function exhibits constant, increasing or decreasing returns to scale.
Codex Input	In Microeconomics, write a program to check if the production functions: $F(L, K) = L^2 * K^{1/2}$ exhibits constant, increasing or decreasing returns to scale. In Microeconomics, write a program to check if the production functions: $F(L, K) = L + L^{1/2} * K^{1/2}$ exhibits constant, increasing or decreasing returns to scale. In Microeconomics, write a program to check if the production functions: $F(L, K) = 2 * L + K$ exhibits constant, increasing or decreasing returns to scale.
Codex Output	<pre> import numpy as np L = np.array([1, 2, 3, 4, 5]) K = np.array([1, 2, 3, 4, 5])  # Define production function def F(L, K):     return L**2 * K**(1/2)  if F(L, K).all() == F(2*L, 2*K).all():     print('Returns to scale are constant') elif F(L, K).all() &lt; F(2*L, 2*K).all():     print('Returns to scale are increasing') else:     print('Returns to scale are decreasing')  # Import packages import numpy as np  # Define functions def F(L, K):     return L + np.sqrt(L) * np.sqrt(K)  # Define parameters L = np.array([1, 2, 3, 4, 5]) K = np.array([1, 2, 3, 4, 5])  if all(F(L, K) == F(L, K)[0]):     print('Constant returns to scale') elif all(F(L, K) &gt; F(L, K)[0]):     print('Increasing returns to scale') elif all(F(L, K) &lt; F(L, K)[0]):     print('Decreasing returns to scale') else:     print('Mixed returns to scale')  import numpy as np L = np.array([1, 2, 3, 4, 5]) K = np.array([1, 2, 3, 4, 5])  # Define production function def F(L, K):     return 2*L + K  if F(L, K) == 2*L + K:     print('Returns to scale are constant') elif F(L, K) &gt; 2*L + K:     print('Returns to scale are increasing') else:     print('Returns to scale are decreasing') </pre>
Solution	IRS, CRS, CRS

Question 7 from MIT 14.01

Topic	Production and Costs
Original Question	<p>You manage a factory that produces cans of peanut butter. The current market price is \$10/can, and you know the following about your costs: <math>MC(5) = 10</math>, <math>ATC(5) = 6</math> <math>MC(4) = 4</math>, <math>ATC(4) = 4</math></p> <p>A case of food poisoning breaks out due to your peanut butter, and you lose a lawsuit against your company. As punishment, Judge Judy decides to take away all of your profits, and considers the following two options to be equivalent: i. Pay a lump sum in the amount of your profits. ii. Impose a tax of <math>\\$[P - ATC(q^*)]</math> per can since that is your current profit per can, where <math>q^*</math> is the profit maximizing output before the lawsuit. Judge Judy gives you the option of choosing either plan. Which plan would you choose? Provide intuition. Hint: a clear diagram may be helpful.</p>
Codex Input	In Principles of Microeconomics, the current market price is \$10/can, and you know the following about your costs: $MC(5) = 10$ , $ATC(5) = 6$ $MC(4) = 4$ , $ATC(4) = 4$ . Write a program to decide: paying a one-time fine in the amount of your current profits or having a tax of $PATC(q_s)$ dollars per can?
Codex Output	<pre># Define the variables MC = [5, 4] ATC = [6, 4] P = 10  # Define the function def profit(MC, ATC, P):     if P - ATC[0] &gt; P - ATC[1]:         print("You should pay a one-time fine in the amount             of your current profits.")     else:         print("You should have a tax of " + str(P - ATC[0])             + " dollars per can.")  # Call the function profit(MC, ATC, P)</pre>
Solution	Choose to take the tax. The tax would be \$4/can. The firm can reduce its quantity to still make a profit.

Question 8 from MIT 14.01

Topic	Supply/Demand
Original Question	Given the utility function: $U(S, C) = 4 * \ln(S) + 6 * \ln(C)$ , compute the marginal rate of substitution of S for C. Is the marginal rate of substitution increasing or decreasing in S.
Codex Input	In Principles of Microeconomics, Given the utility function $U(S, C) = 4 * \ln(S) + 6 * \ln(C)$ , determine whether the marginal rate of substitution increases or decreases as S increases.
Codex Output	<pre># Solution: # # The MRS is defined as: # # \$\$ # \frac{u'(S, C)}{u'(C, S)} = \frac{4}{6} # \$\$ # # As S increases, the numerator increases, and the # denominator decreases. # # Therefore, the MRS decreases as S increases.</pre>
Solution	MRS is decreasing

Question 9 from MIT 14.01

Topic	
Original Question	Suppose the demand for apples is $Q_D = 550 - 50 * P$ and the industry supply curve is $Q_S = -12.5 + 62.5 * P$ . Calculate the equilibrium price and quantity.
Codex Input	In Principles of Microeconomics, Suppose the demand for apples is $Q_D = 550 - 50 * P$ and the industry supply curve is $Q_S = -12.5 + 62.5 * P$ . Write a program to compute the equilibrium price and quantity using the equilibrium price.
Codex Output	<pre> # Import packages import numpy as np import matplotlib.pyplot as plt  # Define demand and supply functions def demand(price):     quantity = 550 - 50*price     return quantity  def supply(price):     quantity = -12.5 + 62.5*price     return quantity  # Define equilibrium function def equilibrium(price):     quantity = demand(price) - supply(price)     return quantity  # Define price range price_range = np.arange(0,10,0.1)  # Plot demand and supply curves plt.plot(price_range, demand(price_range), label = 'Demand') plt.plot(price_range, supply(price_range), label = 'Supply') plt.xlabel('Price') plt.ylabel('Quantity') plt.legend() plt.show()  # Find equilibrium price equilibrium_price = scipy.optimize.fsolve(equilibrium, 1) print('Equilibrium price is', equilibrium_price)  # Find equilibrium quantity equilibrium_quantity = demand(equilibrium_price) print('Equilibrium quantity is', equilibrium_quantity) </pre>
Solution	Price = 5, Quantity = 300

Question 10 from MIT 14.01

Topic	Utility
Original Question	Lauren wants to be a physicist, so she places more weight on her physics test score. Her utility function is given by $u(p, e) = 0.6\ln(p) + 0.4\ln(e)$ . where p is the score on the physics final and e is the score on the economics final. Although she cares more about physics, she is better at economics; for each hour spent studying economics she will increase her score by 3 points, but her physics score will only increase by 2 points for every hour spent studying physics, How many hours should Lauren optimally spend studying physics? How many hour should Lauren study economics?
Codex Input	In Principles of Microeconomics, Given the utility function, $u(x, y) = 0.6 * \text{math.log}(3 * x) + 0.4 * \text{math.log}(2 * y)$ , with the constraints $x+y=24$ , where x is a positive real number and y is a positive real number. Write an optimal program using scipy to find the values of x and y that minimizes u scaled by negative 1.
Codex Output	<pre>import numpy as np from scipy.optimize import minimize  def utility(x):     return -0.6*np.log(3*x[0])-0.4*np.log(2*x[1])  def constraint(x):     return x[0]+x[1]-24  x0 = [1, 1]  con = {'type': 'eq', 'fun': constraint}  sol = minimize(utility, x0, method='SLSQP', constraints=con)  print(sol)</pre>
Solution	Hours studying economics = 9.6, Hours studying physics = 14.4

Question 11 from MIT 14.01

Topic	Trade
Original Question	<p>Consider the production of wine and cheese in France and Spain. This table gives the number of necessary hours to produce each (labor is the only input):</p> <p>France takes 4 hours to produce 1 Kilo of Cheese. Spain takes 6 hours to produce 1 Kilo of Cheese. France takes 6 hours to produce 1 Bottle of wine. Spain takes 12 hours to produce 1 Bottle of wine. For each good, which country has an absolute advantage? For each good, which country has a comparative advantage?</p>
Codex Input	<p>In Principles of Microeconomics, France takes 4 hours to produce 1 Kilo of Cheese. Spain takes 6 hours to produce 1 Kilo of Cheese. France takes 6 hours to produce 1 Bottle of wine. Spain takes 12 hours to produce 1 Bottle of wine. For each good, write a program to compute which country has an absolute advantage.</p> <p>In Principles of Microeconomics, France takes 4 hours to produce 1 Kilo of Cheese. Spain takes 6 hours to produce 1 Kilo of Cheese. France takes 6 hours to produce 1 Bottle of wine. Spain takes 12 hours to produce 1 Bottle of Wine. Compute which country has a comparative advantage for each good.</p>
Codex Output	<pre> class Country:     def __init__(self, name, cheese_time, wine_time):         self.name = name         self.cheese_time = cheese_time         self.wine_time = wine_time  # Create a list of countries countries = [Country("France", 4, 6), Country("Spain",         6, 12)]  # Create a function to compute the absolute advantage def absolute_advantage(country1, country2):     if country1.cheese_time &lt; country2.cheese_time and         country1.wine_time &lt; country2.wine_time:         print(country1.name + " has an absolute advantage in             both goods.")     elif country1.cheese_time &lt; country2.cheese_time:         print(country1.name + " has an absolute advantage in             cheese.")     elif country1.wine_time &lt; country2.wine_time:         print(country1.name + " has an absolute advantage in             wine.")     else:         print(country1.name + " has no absolute advantage.")  # Call the function absolute_advantage(countries[0], countries[1])  # Solution: # # We can use the following table to solve this problem: # #   Good   Production Time (Hours)   #  ----- -----  #   Cheese   4   #   Wine   6   # #   Good   Comparative Advantage   #  ----- -----  #   Cheese   Spain   #   Wine   France   # # We can see that Spain has a comparative advantage in     Cheese, and France has a comparative advantage in wine. </pre>
Solution	<p>France has an absolute advantage in both cheese and wine. Spain has a comparative advantage in cheese. France has a comparative advantage in wine.</p>

Question 12 from MIT 14.01

Topic	Utility
Original Question	Chloe consumes only books (x) and video games (y). Her preferences can be represented by the following utility function: $U(x, y) = x * (y^2)$ . The price of books is $p_x$ , the price of video games is $p_y$ , and Chloe has an income of m dollars. Compute Chloe's budget constraint.
Codex Input	Using SymPy, write a program to find the equation of the budget constraint in terms of $p_x$ , $p_y$ and m. $p_x$ is the price of x, $p_y$ is the price of y, and m is money spent.
Codex Output	<pre># Solution:  # import sympy module from sympy import *  # declare symbols x, y, p_x, p_y, m = symbols('x y p_x p_y m')  # declare equation eq = Eq(p_x * x + p_y * y, m)  # print equation print(eq)</pre>
Solution	Budget constraint is $p_x(x) + p_y(y) \leq m$

Question 13 from MIT 14.01

Topic	Utility
Original Question	Chloe consumes only books (x) and video games (y). Her preferences can be represented by the following utility function: $U(x, y) = x * (y^2)$ . Calculate the Marginal Rate of Substitution (at an arbitrary bundle (x, y)).
Codex Input	Using SymPy, write a program to find the equation of Marginal Rate of Substitution at any point (x,y) given the utility function: $U(x, y) = x * (y^2)$ .
Codex Output	<pre>import sympy as sp  x, y = sp.symbols('x y')  U = x*(y**2)  MRS = (U.diff(x)/U.diff(y))  print(MRS)</pre>
Solution	Marginal Rate of Substitution = $y/2x$

Question 14 from MIT 14.01

Topic	Utility
Original Question	Consider an economy with only one good: food. There are three people in the economy, A, B and C. A has 400 units of food, B has 100 units, and C has only 16 units. All have the same utility, $U_i = \sqrt{f}$ for $i = A, B, C$ . The government decides to redistribute food more equally, so it takes 175 units from A and gives them to B. However, the government spoils 79 of these units in transportation, so B ultimately gets only 96 units of food. What is each person's utility level?
Codex Input	There are three people in the economy, A, B and C. A has 400 units of food, B has 100 units, and C has only 16 units. All have the same utility function, $U_i = \sqrt{f}$ . The government takes 175 units of food from A and gives them to B. However, the government spoils 79 of these units in transportation, so B ultimately gets only 96 units of food. Write a program to compute A's utility level. Write a program to compute B's utility level. Write a program to compute C's utility level.
Codex Output	<pre>def utility(food):     return math.sqrt(food)  def main():     A = 400     B = 100     C = 16     gov_transfer = 175     spoiled = 79     B_food = B + gov_transfer - spoiled     A_utility = utility(A - gov_transfer)     B_utility = utility(B_food)     C_utility = utility(C)     print("A's utility: ", A_utility)     print("B's utility: ", B_utility)     print("C's utility: ", C_utility)  if __name__ == "__main__":     main()</pre>
Solution	A's Utility = 15, B's Utility = 14, C's Utility = 4

Question 15 from MIT 14.01

Topic	Utility
Original Question	A consumer's preferences are represented by the following utility function: $u(x, y) = x^{1/2} + y$ . Obtain the MRS of the consumer at an arbitrary point $(x^*, y^*)$ , where $x^* > 0$ and $y^* > 0$ .
Codex Input	In Principles of Microeconomics, write a program using sympy to obtain the MRS at any point given the utility function: $u(x, y) = x^{1/2} + y$ .
Codex Output	<pre>import sympy as sp  x, y = sp.symbols('x y')  u = x**(1/2) + y  mu = sp.diff(u, x) / sp.diff(u, y)  print(mu)</pre>
Solution	Marginal Rate of Substitution = $(-1/2) * (x^{-1/2})$

Question 16 from MIT 14.01

Topic	Utility
Original Question	Draw the Engel curve for video games. Are video games an inferior or a normal good?
Codex Input	p is the price of video games, and y is the video games consumed, and x is income. The demand function is $y = x/(3 * p)$ Use sympy to find the slope of the engel curve by differentiating $x/(3 * p)$ with respect to x. Find the y-intercept. Tell me if video games are normal or inferior goods.
Codex Output	<pre># Define the variables x, y, p = sympy.symbols('x y p')  # Define the demand function demand_function = x/(3*p)  # Find the slope of the demand function slope = sympy.diff(demand_function, x)  # Find the y-intercept y_intercept = demand_function.subs(x, 0)  # Tell me if video games are normal or inferior goods print(f'The slope is {slope} and the y-intercept is {y_intercept}') print('Video games are normal goods')</pre>
Solution	Video games are a normal good.

Question 17 from MIT 14.01

Topic	Utility
Original Question	Suppose a worker has preferences over consumption and leisure that can be represented by the following utility function: $U = \ln(c) + \ln(l)$ There are 16 hours per day available for leisure (l) and work (L). The hourly wage is w, and assume that the price of each unit of consumption is \$1. Write down the worker's budget constraint in terms of c and L. Find the optimal consumption and work as a function of w.
Codex Input	A utility function is $U = \ln(c) + \ln(l)$ . Create $c = w*h$ . Create $l+h=16$ . Use Sympy to maximize U and create Eq $dU=0$ before solving for and setting to h. Then print c.
Codex Output	<pre>from sympy import *  w, h = symbols('w h')  c = w*h l = 16-h  U = ln(c) + ln(l)  dU = diff(U, h)  h_max = solve(Eq(dU, 0), h)  print(c.subs(h, h_max[0]))</pre>
Solution	Optimal consumption = 8w

Question 18 from MIT 14.01



Topic	Government Regulations
Original Question	Determine whether the following statements are True or False. Explain your answer. A government sets a price ceiling for widgets that is below the equilibrium price. This intervention will always decrease the producer surplus, increase consumer surplus and decrease total surplus.
Codex Input	In Principles of Microeconomics, is the following statement true or false, then explain why. A government sets a price ceiling for widgets that is below the equilibrium price. This intervention will always decrease the producer surplus, increase consumer surplus and decrease total surplus.
Codex Output	# False. The producer surplus will increase, consumer surplus will decrease and total surplus will decrease.
Solution	False, it might increase or decrease consumer surplus. Price decreases but also quantity decreases. The firm(s) might also exit the market if the price is below average cost which would result in zero consumer surplus.

Question 19 from MIT 14.01

Topic	In the short run, a firm has fixed capital K. We know that its short-run cost function is $C(q) = q^3 - (2 * q^2) + (2 * q) + 2$ . Plot the short-run marginal cost and average variable cost as a function of q.
Original Question	Given the short-run total cost function $SRTC(q) = q * 3 - (2 * q * 2) + (2 * q) + 2$ . Remove $SRTC(0)$ to compute $VC(q)$ . Find average variable cost, $AVC(q)$ . Plot $AVC(q)$ . Find marginal cost, $MC(q)$ . Plot $MC(q)$ .
Codex Input	Given the short-run total cost function $SRTC(q) = q**3-(2*q**2)+(2*q)+2$ . Remove $SRTC(0)$ to compute $VC(q)$ . Find average variable cost, $AVC(q)$ . Plot $AVC(q)$ . Find marginal cost, $MC(q)$ . Plot $MC(q)$ .
Codex Output	<pre>import numpy as np import matplotlib.pyplot as plt  def SRTC (q) :     return q**3-(2*q**2)+(2*q)+2  def VC (q) :     return SRTC (q)-SRTC (0)  def AVC (q) :     return VC (q) /q  def MC (q) :     return 3*q**2-4*q+2  def main() :     q = np.linspace(0,10,100)     plt.plot (q,AVC (q))     plt.plot (q,MC (q))     plt.show()  if __name__ == "__main__":     main()</pre>
Solution	$MCSR(q) = 3q^2 - 4q + 2$ , $AVCSR(q) = q^2 - 2q + 2$

Question 20 from MIT 14.01

## D MIT 18.05: Introduction to Probability and Statistics

451

Topic	Probability
Original question	In class we worked with non-transitive dice: Red: 3 3 3 3 3 6; Green: 1 4 4 4 4 4; White: 2 2 2 5 5 5. Find the probability that white beats green.
Codex input	Given non-transitive dice: Red: 3 3 3 3 3 6; Green: 1 4 4 4 4 4; White: 2 2 2 5 5 5, write a program that computes the probability that white beats green.
Codex output	<pre>import itertools  red = [3, 3, 3, 3, 3, 6] green = [1, 4, 4, 4, 4, 4] white = [2, 2, 2, 5, 5, 5]  def prob_white_beats_green(red, green, white):     rolls = list(itertools.product(red, green, white))     wins = [1 for (r, g, w) in rolls if w &gt; g]     return sum(wins) / len(rolls)  prob_white_beats_green(red, green, white)</pre>
Solution	7/12

Question 1 from 18.05

Topic	Probability
Original question	Calculate the probability of getting a two-pair poker hand.
Codex input	<p>A hand is a set of 5 cards that are drawn randomly from a standard 52 card deck with 13 ranks of 4 cards each.</p> <p>A two-pair poker hand is a hand that contains 3 unique ranks, where no more than 2 cards in the hand can share the same rank. That is, 3 or more cards cannot share the same rank.</p> <p>Write a program that generates simulations for calculating the average probability of getting a two-pair poker hand.</p>
Codex output	<pre>import random  def get_rank_counts(hand):     """     Returns a dictionary with ranks as keys and the number     of cards with that rank as values.     """     rank_counts = {}     for card in hand:         rank = card[0]         if rank in rank_counts:             rank_counts[rank] += 1         else:             rank_counts[rank] = 1     return rank_counts  def has_two_pair(hand):     """     Returns True if the hand has two pairs, and False     otherwise.     """     rank_counts = get_rank_counts(hand)     two_pair = False     for rank in rank_counts:         if rank_counts[rank] == 2:             if two_pair:                 return True             else:                 two_pair = True     return False  def run_simulation():     """     Runs a simulation of drawing 5 cards randomly from a     deck of 52 cards.     """     two_pair = 0     for i in range(1000000):         deck = [r+s for r in '23456789TJQKA' for s in 'SHDC']         random.shuffle(deck)         hand = deck[:5]         if has_two_pair(hand):             two_pair += 1     return two_pair/1000000  print(run_simulation())</pre>
Solution	0.047539

Question 2 from 18.05

Topic	Probability
Original question	Calculate the probability of getting a three-of-a-kind poker hand.
Codex input	<p>A hand is a set of 5 cards that are drawn randomly from a standard 52 card deck with 13 ranks of 4 cards each.</p> <p>A three-pair poker hand is a hand that contains exactly 3 cards that share one rank, 1 card of another rank, and a final card of another rank.</p> <p>Write a program that generates simulations for calculating the average probability of getting a three-pair poker hand.</p>
Codex output	<pre>import random  def get_rank_counts(hand):     """     Return a dictionary with ranks as keys and their counts     as values.     """     ranks = [card[0] for card in hand]     return {rank: ranks.count(rank) for rank in ranks}  def is_three_pair(hand):     """     Return True if hand is a three pair hand.     """     rank_counts = get_rank_counts(hand)     return 3 in rank_counts.values()  def get_three_pair_probability(n):     """     Return the probability of getting a three pair hand.     """     three_pair_count = 0     for _ in range(n):         deck = [             '2C', '2D', '2H', '2S',             '3C', '3D', '3H', '3S',             '4C', '4D', '4H', '4S',             '5C', '5D', '5H', '5S',             '6C', '6D', '6H', '6S',             '7C', '7D', '7H', '7S',             '8C', '8D', '8H', '8S',             '9C', '9D', '9H', '9S',             'TC', 'TD', 'TH', 'TS',             'JC', 'JD', 'JH', 'JS',             'QC', 'QD', 'QH', 'QS',             'KC', 'KD', 'KH', 'KS',             'AC', 'AD', 'AH', 'AS'         ]         hand = random.sample(deck, 5)         if is_three_pair(hand):             three_pair_count += 1     return three_pair_count / n  print(get_three_pair_probability(1000000))</pre>
Solution	0.021128

Question 3 from 18.05

Topic	Probability
Original question	In class we worked with non-transitive dice: Red: 3 3 3 3 6; Green: 1 4 4 4 4; White: 2 2 2 5 5. Find the probability that green beats red.
Codex input	Given two non-transitive dice: Red: 3 3 3 3 6; Green: 1 4 4 4 4. Write a program that computes the probability that green beats red.
Codex output	<pre>def dice_prob(dice1, dice2):     """     dice1: list of dice 1     dice2: list of dice 2     """     # Your code here     red_wins = 0     green_wins = 0     for i in dice1:         for j in dice2:             if i &gt; j:                 red_wins += 1             elif j &gt; i:                 green_wins += 1     return green_wins / (red_wins + green_wins)  # Test Cases dice1 = [3, 3, 3, 3, 6] dice2 = [1, 4, 4, 4, 4] print(dice_prob(dice1, dice2))</pre>
Solution	25/36

Question 4 from 18.05

Topic	Conditional Probability
Original question	Mr. Smith has two children. At least one of them is a boy. What is the probability that both children are boys?
Codex input	Write a program that computes the probability that both children are boys given that one of them is a boy.
Codex output	<pre># probability that one child is a boy p_one_boy = 1/4  # probability that both children are boys p_both_boys = 1/4  # probability that one of them is a boy p_one_is_boy = 3/4  # probability that both children are boys given that one of   them is a boy p_both_boys_given_one_boy = p_both_boys / p_one_is_boy  print(p_both_boys_given_one_boy)</pre>
Solution	1/3

Question 5 from 18.05

Topic	Discrete Random Variables, Standard Deviation
Original question	Let X be the result of rolling a fair 4-sided die. Let Y be the result of rolling a fair 6-sided die. Let Z be the average of X and Y. Find the standard deviation of X.
Codex input	In Probability, Write a program that computes the variance and standard deviation of rolling a fair 4-sided die 1000000 times.
Codex output	<pre>import random  def variance(lst):     mean = sum(lst) / len(lst)     return sum((i - mean) ** 2 for i in lst) / len(lst)  def standard_deviation(lst):     return variance(lst) ** 0.5  def main():     lst = []     for i in range(1000000):         lst.append(random.randint(1, 4))     print(variance(lst))     print(standard_deviation(lst))  if __name__ == '__main__':     main()</pre>
Solution	1.118

Question 6 from 18.05

Topic	Discrete Random Variables, Standard Deviation
Original question	Let X be the result of rolling a fair 4-sided die. Let Y be the result of rolling a fair 6-sided die. Let Z be the average of X and Y. Find the standard deviation of Y.
Codex input	In Probability, Write a program that computes the variance and standard deviation of rolling a fair 6-sided die 1000000 times.
Codex output	<pre>import statistics  def variance_and_standard_deviation():     data = []     for i in range(1000000):         data.append(random.randint(1, 6))     print(statistics.variance(data))     print(statistics.stdev(data))  variance_and_standard_deviation()</pre>
Solution	1.708

Question 7 from 18.05

Topic	Discrete Random Variables, Standard Deviation
Original question	Let X be the result of rolling a fair 4-sided die. Let Y be the result of rolling a fair 6-sided die. Let Z be the average of X and Y. Find the standard deviation of Z.
Codex input	Let X be the result of rolling a fair 4-sided die. Let Y be the result of rolling a fair 6-sided die. Let Z be the average of X and Y. Write a program that computes the standard deviation of Z.
Codex output	<pre> import math  def standard_deviation(lst):     mean = sum(lst) / len(lst)     return math.sqrt(sum([(x - mean) ** 2 for x in lst]) /                        len(lst))  def main():     X = [1, 2, 3, 4]     Y = [1, 2, 3, 4, 5, 6]     Z = []     for i in X:         for j in Y:             Z.append((i + j) / 2)     print(standard_deviation(Z))  if __name__ == "__main__":     main() </pre>
Solution	1.021

Question 8 from 18.05

Topic	Discrete Random Variables, Expectation
Original question	Let $X$ be the result of rolling a fair 4-sided die. Let $Y$ be the result of rolling a fair 6-sided die. You win $2X$ dollars if $X > Y$ and lose 1 dollar otherwise. After playing this game 60 times, what is your expected total gain?
Codex input	In Probability, $X$ is the result of rolling a fair 4-sided die and $Y$ is the result of rolling a fair 6-sided die. You win $2X$ if $X > Y$ and lose 1 otherwise. Write a program that computes the expected total gain after rolling both dice 60 times.
Codex output	<pre>import random  def roll_dice():     return random.randint(1, 4)  def roll_dice_6():     return random.randint(1, 6)  def play_game():     total_gain = 0     for i in range(60):         x = roll_dice()         y = roll_dice_6()         if x &gt; y:             total_gain += 2 * x         else:             total_gain -= 1     return total_gain  def main():     total_gain = 0     for i in range(1000):         total_gain += play_game()     print(total_gain / 1000)  if __name__ == '__main__':     main()</pre>
Solution	55

Question 9 from 18.05

Topic	Counting, Continuous Variables
Original question	Boxes of Raisin Bran cereal are 30cm tall. Due to settling, boxes have a higher density of raisins at the bottom ( $h = 0$ ) than at the top ( $h = 30$ ). Suppose the density (in raisins per cm of height) is given by $f(h) = 40 - h$ . How many raisins are in a box?
Codex input	Boxes of Raisin Bran cereal are 30cm tall. Suppose the density of raisins per cm of height is given by $f(h) = 40 - h$ . Using <code>scipy.integrate</code> , write a program that computes how many raisins are in a box.
Codex output	<pre>import numpy as np import scipy.integrate as integrate  def f(h):     return 40 - h  def integrate_f(a, b):     return integrate.quad(f, a, b)  print(integrate_f(0, 30))</pre>
Solution	750

Question 10 from 18.05



Topic	Probability, Continuous Variables
Original question	Boxes of Raisin Bran cereal are 30cm tall. Due to settling, boxes have a higher density of raisins at the bottom ( $h = 0$ ) than at the top ( $h = 30$ ). Suppose the density (in raisins per cm of height) is given by $f(h) = 40 - h$ . What is the probability that a random raisin is in the bottom third of the box?
Codex input	$h$ ranges from 0 to 30. Integrate $f(h) = 40 - h$ from 0 to 10. Divide by the integration of $h$ from 0 to 30.
Codex output	<pre>def f(h):     return 40 - h  def integrate(f, a, b, N):     """     Integrates f from a to b using N intervals     """     dx = (b - a) / N     total = 0     for i in range(N):         total += f(a + i * dx)     return total * dx  print(integrate(f, 0, 10, 100000) / integrate(f, 0, 30, 100000))</pre>
Solution	7/15

#### Question 11 from 18.05

Topic	Prior Odds
Original question	You have a drawer that contains 50 coins. 10 coins have probability $p = 0.3$ of heads, 30 coins have probability $p = 0.5$ and 10 coins have probability $p = 0.7$ . You pick one coin at random from the drawer and flip it. What are the (prior) odds you chose a 0.3 coin?
Codex input	You have a drawer that contains 10 white coins and 40 black coins. You pick one coin at random from the drawer. Write a program that computes the odds that you chose a white coin.  In probability, the odds of an event $E$ is $P(E)/P(E^c)$ .
Codex output	<pre># P(E) = 10/50 # P(E^c) = 40/50 # odds = P(E)/P(E^c) = 10/40  print(10/40)</pre>
Solution	1/4

#### Question 12 from 18.05

Topic	Joint Distributions
Original question	Suppose $X$ and $Y$ have joint pdf $f(x, y) = c(x^2 + xy)$ on $[0, 1] \times [0, 1]$ . Find $c$ .
Codex input	Define variables $x, y, c$ . Using Sympy, integrate $f(x, y) = c(x^2 + xy)$ from 0 to 1 for $x$ and 0 to 1 for $y$ ; set it equal to 1 in an equation. Find $c$ that makes the expression 1.
Codex output	<pre>import sympy as sp  x, y, c = sp.symbols('x y c')  f = c * (x**2 + x*y)  sp.integrate(f, (x, 0, 1), (y, 0, 1))  sp.solve(sp.Eq(1, sp.integrate(f, (x, 0, 1), (y, 0, 1))), c)</pre>
Solution	12/7

#### Question 13 from 18.05

Topic	Central Limit Theorem, Probability
Original question	To head the newly formed US Dept. of Statistics, suppose that 50% of the population supports Erika, 20% supports Ruthi, and the rest are split between Peter, Jon and Jerry. A poll asks 400 random people who they support. Use the central limit theorem to estimate the probability that at least 52.5% of those polled prefer Erika?
Codex input	Suppose that 50% of the population supports Erika. A poll asks 400 random people who they support. Write a program that uses the central limit theorem to compute the probability that at least 52.5% of those polled support Erika.
Codex output	<pre> import random  def poll(n):     """     n: number of people polled     returns: percentage of people in favor of Erika     """     # generate a list of n random numbers between 0 and 1     # if the number is less than 0.5, the person is in favor     # of Erika     # if the number is greater than 0.5, the person is in     # favor of Erica     # return the percentage of people in favor of Erika     count = 0     for i in range(n):         if random.random() &lt; 0.5:             count += 1     return count / n  def poll_prob(n, m):     """     n: number of people polled     m: number of times the poll is conducted     returns: probability that at least 52.5% of the     population supports Erika     """     # run poll_prob() m times     # count the number of times the poll is in favor of Erika     # return the number of times the poll is in favor of     # Erika divided by m     count = 0     for i in range(m):         if poll(n) &gt;= 0.525:             count += 1     return count / m  print(poll_prob(400, 10000)) </pre>
Solution	0.16

Question 14 from 18.05

Topic	Central Limit Theorem, Probability
Original question	To head the newly formed US Dept. of Statistics, suppose that 50% of the population supports Erika, 20% supports Ruthi, and the rest are split between Peter, Jon and Jerry. A poll asks 400 random people who they support. Use the central limit theorem to estimate the probability that less than 25% of those polled prefer Peter, Jon or Jerry?
Codex input	Suppose that 30% of the population supports Peter. A poll asks 400 random people who they support. Write a program that uses the central limit theorem to compute the probability that less than 25% of those polled support Peter.
Codex output	<pre>import random  def poll(n, p):     count = 0     for i in range(n):         if random.random() &lt; p:             count += 1     return count  def main():     peter_support = 0.3     n = 400     p = 0.25     poll_size = 1000     count = 0     for i in range(poll_size):         if poll(n, peter_support) &lt; p * n:             count += 1     print(count / poll_size)  if __name__ == "__main__":     main()</pre>
Solution	0.0145

Question 15 from 18.05

Topic	Uniform Random Variables, Probability
Original question	Alice and Bob are trying to meet for lunch and both will arrive, independently of each other, uniformly and at random between noon and 1pm. Let A and B be the number of minutes after noon at which Alice and Bob arrive, respectively. Then A and B are independent uniformly distributed random variables on [0, 60]. Find the probability that Alice arrives before 12:30.
Codex input	A and B are trying to meet for lunch between 12:00 and 1:00. Alice and Bob arriving for lunch are independent and uniformly distributed events. Write a function that finds the probability that A arrives before 12:30.
Codex output	<pre>def prob_a_before_1230(a, b):     return a / (a + b)  print(prob_a_before_1230(0.5, 0.5))</pre>
Solution	1/2

Question 16 from 18.05

Topic	Uniform Random Variables, Probability
Original question	Alice and Bob are trying to meet for lunch and both will arrive, independently of each other, uniformly and at random between noon and 1pm. Let A and B be the number of minutes after noon at which Alice and Bob arrive, respectively. Then A and B are independent uniformly distributed random variables on [0, 60]. Now suppose that Alice and Bob are both rather impatient and will leave if they have to wait more than 15 minutes for the other to arrive. What is the probability that Alice and Bob will have lunch together?
Codex input	Alice and Bob are trying to meet for lunch and both will arrive, independently of each other, uniformly and at random between 12pm and 1pm. Suppose that Alice and Bob are both rather impatient and will leave if they have to wait more than 15 minutes for the other to arrive. Write a function that randomly computes the probability that Alice and Bob will have lunch together.
Codex output	<pre>import random  def lunch_together():     # Alice arrives at time     time_a = random.uniform(0, 60)     # Bob arrives at time     time_b = random.uniform(0, 60)     # If both arrive within 15 minutes of each other, they     # have lunch together     if abs(time_a - time_b) &lt;= 15:         return True     else:         return False  def prob_lunch_together():     # Run lunch_together() 10,000 times     count = 0     for i in range(10000):         if lunch_together():             count += 1     # Return probability     return count / 10000  print(prob_lunch_together())</pre>
Solution	7/16

Question 17 from 18.05

Topic	Prior Odds
Original question	You have a drawer that contains 50 coins. 10 coins have probability $p = 0.3$ of heads, 30 coins have probability $p = 0.5$ and 10 coins have probability $p = 0.7$ . You pick one coin at random from the drawer and flip it. What are the (prior) odds you chose a 0.7 coin?
Codex input	You have a drawer that contains 10 white coins and 40 black coins. You pick one coin at random from the drawer. Write a program that computes the odds that you chose a white coin.  In probability, the odds of an event E is $P(E)/P(E^c)$ .
Codex output	<pre># P(E) = 10/50 # P(E^c) = 40/50 # odds = P(E) / P(E^c) = 10/40  print(10/40)</pre>
Solution	1/4

Question 18 from 18.05

Topic	Null Hypothesis Significance Testing, Type I Error
Original question	One generates a number $x$ from a uniform distribution on the interval $[0, \theta]$ . One decides to test $H_0: \theta = 2$ against $H_A: \theta = 2$ by rejecting $H_0$ if $x \leq 0.1$ or $x \geq 1.9$ . Compute the probability of a type I error.
Codex input	One generates a number $x$ from a uniform distribution on the interval $[0, \theta]$ . One decides to test $H_0: \theta = 2$ against $H_A: \theta = 2$ by rejecting $H_0$ if $x \leq 0.1$ or $x \geq 1.9$ . Using numpy, write a program that computes the probability of a type I error.
Codex output	<pre>import numpy as np  theta = 2 n = 100000  x = np.random.uniform(0, theta, n)  type_I_error = np.sum(x &lt;= 0.1) + np.sum(x &gt;= 1.9)  print(type_I_error / n)</pre>
Solution	0.1

Question 19 from 18.05

Topic	Null Hypothesis Significance Testing, Type II Error
Original question	One generates a number $x$ from a uniform distribution on the interval $[0, \theta]$ . One decides to test $H_0: \theta = 2$ against $H_A: \theta = 2$ by rejecting $H_0$ if $x \leq 0.1$ or $x \geq 1.9$ . Compute the probability of a type II error if the true value of $\theta$ is 2.5.
Codex input	One generates a number $x$ from a uniform distribution on the interval $[0, \theta]$ .  One decides to test $H_0: \theta = 2$ against $H_A: \theta = 2$ by rejecting $H_0$ if $x \leq 0.1$ or $x \geq 1.9$ , and accepted otherwise.  Given that $\theta = 2.5$ , write a program that computes the type II error.  The type II error is the probability that the null hypothesis is mistakenly accepted.
Codex output	<pre>import numpy as np  theta = 2.5  def type_II_error(theta, x):     return 1 - np.sum(np.logical_or(x &lt; 0.1, x &gt; 1.9)) /         len(x)  x = np.random.uniform(0, theta, 100000)  print(type_II_error(theta, x))</pre>
Solution	0.72

Question 20 from 18.05

Topic	Vectors and Linear Combinations (1.1, Q3)
Original question	If $v+w = (5,1)$ and $v-w=(1,5)$ , compute and draw the vectors $v$ and $w$
Codex input	$v$ and $w$ are 1 by2 vectors such that $v + w = (5,1)$ and $v - w = (1,5)$ Write a program to find the vectors $v$ and $w$ Draw the vectors $v$ and $w$
Codex output	<pre> # Solution # # We can solve this problem by using the following steps: # # 1. We know that \$v + w = (5,1)\$ and \$v - w = (1,5)\$ # 2. We can subtract \$v - w\$ from both sides to get \$v + w - (v-w) = (5,1) + (1,5) - ((1,5)-(5,1))\$ # 3. This gives us \$2v = (6,6)\$ and hence \$v = (3,3)\$ # 4. Similarly we can find out that \$w = (-2,-2)\$  import matplotlib.pyplot as plt import numpy as np  plt.quiver(0, 0, v[0], v[1], angles='xy', scale_units='xy',           scale=1) plt.quiver(0, 0, w[0], w[1], angles='xy', scale_units='xy',           scale=1)  plt.xlim(-10, 10) plt.ylim(-10, 10) </pre>
Solution	This problem gives the diagonals $v + w$ and $v - w$ of the parallelogram and asks for the sides: The opposite of Problem 2. In this example $v = (3, 3)$ and $w = (2, -2)$ .

MIT 18.06, Question 1: Original question, Codex input and output, and solution.

Topic	Vectors and Linear Combinations (1.1, Q11)
Original question	Four of the eight corners of a unit cube are $(0,0,0)$ , $(1,0,0)$ , $(0,1,0)$ , $(0,0,1)$ . Find the coordinates of the center point of the cube.
Codex input	Four of the eight corners of a unit cube are $(0,0,0)$ , $(1,0,0)$ , $(0,1,0)$ , $(0,0,1)$ . Find the coordinates of the center point of the cube.
Codex output	<pre> def find_center(x,y,z):     return (x+0.5, y+0.5, z+0.5)  print(find_center(0,0,0)) </pre>
Solution	$(0.5, 0.5, 0.5)$

MIT 18.06, Question 2: Original question, Codex input and output, and solution.

Topic	Vectors and Linear Algebra (1.2 Q28)
Original question	If $v = (1,2)$ draw all vectors $w = (x,y)$ in the $xy$ plane with $\text{dot}(v,w) = x + 2y = 5$ . Why do those $w$ 's lie along a line? Which is the shortest $w$ ?
Codex input	If $v = (1,2)$ draw all vectors $w = (x,y)$ in the $xy$ plane with $\text{dot}(v,w) = x + 2y = 5$ . Why do those $w$ 's lie along a line? Which is the shortest $w$ ?
Codex output	<pre>import matplotlib.pyplot as plt import numpy as np  x = np.linspace(-10, 10, 100) y = 5 - x/2 plt.plot(x, y) plt.show()  plt.arrow(0, 0, 1, 2) # (x1, y1), (x2, y2) plt.show()</pre>
Solution	(1, 2)

MIT 18.06, Question 3: Original question, Codex input and output, and solution.

Topic	Vectors and Linear Algebra (1.3 Q1)
Original question	Find the linear combination $3s_1 + 4s_2 + 5s_3 = b$ . Then write $b$ as a matrix-vector multiplication $Sx$ , with 3, 4, 5 in $x$ . Compute the three dot products (row of $S$ ) $x$ : $s_1 = [1;1;1]$ , $s_2 = [0;1;1]$ , $s_3 = [0;0;1]$
Codex input	Write a function to calculate the linear combination $3*s_1 + 4*s_2 + 5*s_3$ . Let $s_1 = [1;1;1]$ , $s_2 = [0;1;1]$ , $s_3 = [0;0;1]$
Codex output	<pre>def linear_combination(s1, s2, s3):     return 3*s1 + 4*s2 + 5*s3  s1 = np.array([1,1,1]) s2 = np.array([0,1,1]) s3 = np.array([0,0,1])  print(linear_combination(s1, s2, s3))</pre>
Solution	$3s_1 + 4s_2 + 5s_3 = (3, 7, 12)$ . The same vector $b$ comes from $S$ times $x = (3, 4, 5)$ .

MIT 18.06, Question 4: Original question, Codex input and output, and solution.

Topic	Vectors and Linear Algebra (1.3, Q4)
Original question	Find a combination $x_1w_1 + x_2w_2 + x_3w_3$ that gives the zero vector with $x_1 = 1$ $w_1$ is the vector (1;2;3) $w_2$ is the vector (4;5;6) $w_3$ is the vector (7;8;9)
Codex input	Write a program to find a combination $y$ and $z$ such that multiplying $\text{np.array}([1, y, z]).T$ with $\text{np.array}([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) = 0$ . Use sympy and numpy.
Codex output	<pre>import sympy as sp import numpy as np  y = sp.Symbol('y') z = sp.Symbol('z')  eq = np.array([1, y, z]).T @ np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])  sp.solve(eq, [y, z])</pre>
Solution	The combination $0w_1 + 0w_2 + 0w_3$ always gives the zero vector, but this problem looks for other zero combinations (then the vectors are dependent, they lie in a plane): $w_2 = (w_1 + w_3)/2$ so one combination that gives zero is $w_1 - 2w_2 + w_3 = 0$ .

MIT 18.06, Question 5: Original question, Codex input and output, and solution.

Topic	Vectors and Linear Equations (2.1, Q21)
Original question	What 2 by 2 matrix R rotates every vector through 45 degrees? Example: the vector [1,0] goes to $[\sqrt{2}/2, \sqrt{2}/2]$ .
Codex input	What 2 by 2 matrix R rotates every vector through 45 degrees? Example: the vector [1,0] goes to $[\sqrt{2}/2, \sqrt{2}/2]$ .
Codex output	<pre>def rotate_45(vector):     return np.matmul(np.array([[np.sqrt(2)/2, np.sqrt(2)/2],                                [-np.sqrt(2)/2, np.sqrt(2)/2]]), vector)  print(rotate_45([1,0]))</pre>
Solution	$R = (1/2)[\sqrt{2}, -\sqrt{2}; \sqrt{2}, \sqrt{2}]$

MIT 18.06, Question 6: Original question, Codex input and output, and solution.

Topic	Elimination Using Matrices (2.3, Q17)
Original question	The parabola $y = a + bx + cx^2$ goes through the points $(x,y) = (1,4)$ and $(2,8)$ and $(3,14)$ . Find and solve a matrix equation for the unknowns $(a,b,c)$
Codex input	The parabola $y = a + b*x + c*x^2$ goes through the points $(x1,y1) = (1,4)$ and $(x2,y2) = (2,8)$ and $(x3,y3) = (3,14)$ . $y1 = a + b*x1 + c*x1^2$ $y2 = a + b*x2 + c*x2^2$ $y3 = a + b*x3 + c*x3^2$ Solve for $(a, b, c)$
Codex output	<pre>import numpy as np  x1 = 1 x2 = 2 x3 = 3 y1 = 4 y2 = 8 y3 = 14  A = np.array([[1, x1, x1**2], [1, x2, x2**2], [1, x3, x3**2]]) b = np.array([y1, y2, y3])  print(np.linalg.solve(A, b))</pre>
Solution	$a = 2, b = 1, \text{ and } c = 1.$

MIT 18.06, Question 7: Original question, Codex input and output, and solution.



Topic	Rules for Matrix Operations (2.4, Q23b)
Original question	Find a matrix that has $A^2$ does not equal 0 but $A^3 = 0$
Codex input	Loop through random integer matrices with 0 until matrix A is found. A must satisfy conditions: $A**2$ not equals $np.zeros((2,2))$ and $A*3$ equals $np.zeros((2,2))$ .
Codex output	<pre>import numpy as np  def find_matrix_A(n):     """     Find matrix A with nxn dimensions.     """     while True:         A = np.random.randint(0,10, (n,n))         if np.array_equal(A**2, np.zeros((n,n))) == False and             np.array_equal(A*3, np.zeros((n,n))) == True:             return A  def main():     """     Main function.     """     n = int(input("Enter n: "))     A = find_matrix_A(n)     print(A)  if __name__ == "__main__":     main()</pre>
Solution	$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

MIT 18.06, Question 8: Original question, Codex input and output, and solution.

Topic	Rules for Matrix Operations (2.5, Q1)
Original question	Find the inverses (directly or from the 2 by 2 formula) of A, B, C : $A = \begin{bmatrix} 0 & 4 \\ 3 & 0 \end{bmatrix}$ , $B = \begin{bmatrix} 2 & 0 \\ 4 & 2 \end{bmatrix}$ , and $C = \begin{bmatrix} 3 & 4 \\ 5 & 7 \end{bmatrix}$ .
Codex input	Find the inverses of the matrices A, B, C : $A = \begin{bmatrix} 0 & 4 \\ 3 & 0 \end{bmatrix}$ , $B = \begin{bmatrix} 2 & 0 \\ 4 & 2 \end{bmatrix}$ , and $C = \begin{bmatrix} 3 & 4 \\ 5 & 7 \end{bmatrix}$ .
Codex output	<pre>import numpy as np  A = np.array([[0, 4], [3, 0]]) B = np.array([[2, 0], [4, 2]]) C = np.array([[3, 4], [5, 7]])  print(np.linalg.inv(A)) print(np.linalg.inv(B)) print(np.linalg.inv(C))</pre>
Solution	$inv(A) = \begin{bmatrix} 0 & 1/4 \\ 1/3 & 0 \end{bmatrix}$ , $inv(B) = \begin{bmatrix} 1/2 & 0 \\ -1 & 1/2 \end{bmatrix}$ , and $inv(C) = \begin{bmatrix} 7 & -4 \\ -5 & 3 \end{bmatrix}$ .

MIT 18.06, Question 9: Original question, Codex input and output, and solution.

Topic	Rules for Matrix Operations (2.5, Q3)
Original question	Solve for the first column (x, y) and second column (t, z) of $A^{-1}$ : $[10,20;20,50]@x;y = [1;0]$ and $[10,20;20,50]@t;z = [0;1]$ .
Codex input	Solve for the first column (x, y) and second column (t, z) of $A^{-1}$ : $[10,20;20,50]@x;y = [1;0]$ and $[10,20;20,50]@t;z = [0;1]$ .
Codex output	<pre>import numpy as np  A = np.array([[10, 20], [20, 50]]) b = np.array([[1], [0]])  x = np.linalg.inv(A)@b  print(x)  A = np.array([[10, 20], [20, 50]]) b = np.array([[0], [1]])  x = np.linalg.inv(A)@b  print(x)</pre>
Solution	$[x;y] = [0.5;-0.2]$ and $[t;z]=[-0.2;0.1]$ so $\text{inv}(A) = 1/10*[5, -2;-2, 1]$

MIT 18.06, Question 10: Original question, Codex input and output, and solution.

Topic	Inverse Matrices (2.5, Q11b)
Original question	Find singular matrices A and B such that A+B is invertible.
Codex input	Find singular matrices A and B such that A+B is invertible.
Codex output	<pre>import numpy as np import numpy.linalg as la  def find_singular_matrices(n):     """     Find two matrices A and B such that A+B is invertible.     """     A = np.random.rand(n,n)     B = np.random.rand(n,n)     while la.det(A+B) == 0:         A = np.random.rand(n,n)         B = np.random.rand(n,n)     return A,B  def main():     """     Test the find_singular_matrices function.     """     A,B = find_singular_matrices(3)     print(A)     print(B)     print(A+B)  if __name__ == "__main__":     main()</pre>
Solution	$A = [1, 0; 0, 0]$ and $B = [0, 0; 0, 1]$

MIT 18.06, Question 11: Original question, Codex input and output, and solution.

Topic	Nullspaces (3.2, Q20)
Original question	Construct a 2 by 2 matrix whose nullspace equals its column space. This is possible.
Codex input	Write a function that checks the nullspace of a matrix. Write a function that checks the column space of a matrix. Write a function to iteratively search for a 2 by 2 matrix <code>np.array([[a, b], [c, d]])</code> and returns the values of the matrix whose nullspace equals its column space.
Codex output	<pre>def check_nullspace(matrix):     """     Checks the nullspace of a matrix.     """     return np.linalg.matrix_rank(matrix)  def check_columnspace(matrix):     """     Checks the column space of a matrix.     """     return np.linalg.matrix_rank(matrix.T)  def iterative_search(matrix):     """     Iteratively searches for a 2 by 2 matrix np.array([[a,     b], [c, d]]) and returns the values of the matrix     whose nullspace equals its column space.     """     for a in range(1, 10):         for b in range(1, 10):             for c in range(1, 10):                 for d in range(1, 10):                     if check_nullspace(np.array([[a, b], [c,                     d]])) == check_columnspace(np.array([[a,                     b], [c, d]])):                         return np.array([[a, b], [c, d]])  print(iterative_search(np.array([[1, 2], [3, 4]])))</pre>
Solution	$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

MIT 18.06, Question 12: Original question, Codex input and output, and solution.

Topic	Independent, Basis and Dimension (3.4, Q1)
Original question	Show that $v_1, v_2, v_3$ are independent but $v_1, v_2, v_3, v_4$ are dependent. $v_1 = [1;0;0]$ , $v_2 = [1;1;0]$ , $v_3 = [1;1;1]$ , $v_4 = [2;3;4]$
Codex input	Let $v_1 = [1;0;0]$ , $v_2 = [1;1;0]$ , $v_3 = [1;1;1]$ , $v_4 = [2;3;4]$ . Write a function to show that $v_1, v_2,$ and $v_3$ are linearly independent. In linear algebra write a function that checks if vectors $v_1, v_2, v_3, v_4$ are independent. $v_1 = (1,0,0)$ , $v_2 = (1,1,0)$ , $v_3 = (1,1,1)$ , $v_4 = (2,3,4)$ Use arrays for the vectors $v_1, v_2, v_3, v_4$ are check if they are independent
Codex output	<pre>def lin_indep(v1, v2, v3):     return np.linalg.det(np.array([v1, v2, v3])) != 0  v1 = np.array([1, 0, 0]) v2 = np.array([1, 1, 0]) v3 = np.array([1, 1, 1]) v4 = np.array([2, 3, 4])  print(lin_indep(v1, v2, v3))  def is_independent(v1, v2, v3, v4):     return not any([v1.dot(v2), v1.dot(v3), v1.dot(v4),                    v2.dot(v3), v2.dot(v4), v3.dot(v4)])  v1 = np.array([1,0,0]) v2 = np.array([1,1,0]) v3 = np.array([1,1,1]) v4 = np.array([2,3,4])  is_independent(v1, v2, v3, v4)</pre>
Solution	$[1, 1, 1; 0, 1, 1; 0, 0, 1] * [c_1; c_2; c_3] = 0$ gives $c_3 = c_2 = c_1 = 0$ . So those 3 column vectors are independent. But $[1 \ 1 \ 1 \ 2; 0 \ 1 \ 1 \ 3; 0 \ 0 \ 1 \ 4] * c = [0; 0; 0]$ is solved by $c = (1, 1, -4, 1)$ then $v_1 + v_2 - 4v_3 + v_4 = 0$ (dependent)

MIT 18.06, Question 13: Original question, Codex input and output, and solution.

Topic	Orthogonality of the Four Subspaces (4.1, Q25)
Original question	Find $A^T A$ if the columns of $A$ are unit vectors, all mutually perpendicular.
Codex input	Find $A^T A$ if the columns of $A$ are unit vectors, all mutually perpendicular.
Codex output	<pre>import numpy as np  def transpose_multiply(A):     return np.dot(A.T, A)  def main():     A = np.array([[1, 0, 0], [0, 1, 0], [0, 0, 1]])     print(transpose_multiply(A))  if __name__ == "__main__":     main()</pre>
Solution	I

MIT 18.06, Question 14: Original question, Codex input and output, and solution.

Topic	Projections (4.2, Q2b)
Original question	Draw the projection of b onto a: $b=[1;1]$ and $a=[1;-1]$ .
Codex input	The vector b is $[1;1]$ The vector a is $[1;-1]$ Plot the projection of b onto a Plot the projection with circle marker
Codex output	<pre> ##### The vector b is [1;1] The vector a is [1;-1] Plot the projection of b onto a #####  import numpy as np import matplotlib.pyplot as plt  a = np.array([1, -1]) b = np.array([1, 1])  # Projection of b onto a proj_b_a = (np.dot(b, a) / np.dot(a, a)) * a  # Plot plt.plot([0, a[0]], [0, a[1]], 'r', label='a') plt.plot([0, b[0]], [0, b[1]], 'g', label='b') plt.plot([0, proj_b_a[0]], [0, proj_b_a[1]], 'b',          label='projection of b onto a') plt.axis('equal') plt.legend() plt.grid() plt.show()  ##### Plot the projection with circle marker #####  # Plot plt.plot([0, a[0]], [0, a[1]], 'r', label='a') plt.plot([0, b[0]], [0, b[1]], 'g', label='b') plt.plot([0, proj_b_a[0]], [0, proj_b_a[1]], 'bo',          label='projection of b onto a') plt.axis('equal') plt.legend() plt.grid() plt.show() </pre>
Solution	(0,0) with a corresponding plot.

MIT 18.06, Question 15: Original question, Codex input and output, and solution.

Topic	Projections (4.2, Q16)
Original question	What linear combination of (1,2,-1) and (1,0,1) is closest to b=(2,1,1)?
Codex input	Use scipy to optimize for x and y such that $\text{np.array}([1,2,-1]) * x + \text{np.array}([1,0,1]) * y$ is closest to $b = \text{np.array}([2,1,1])$ .
Codex output	<pre>import numpy as np from scipy.optimize import minimize  def objective(x):     return np.sum(np.square(np.array([1,2,-1]) * x[0] +         np.array([1,0,1]) * x[1] - np.array([2,1,1])))  x0 = np.array([0,0])  bnds = ((0, None), (0, None))  res = minimize(objective, x0, method='SLSQP', bounds=bnds)  print(res.x)</pre>
Solution	(2, 1, 1)

MIT 18.06, Question 16: Original question, Codex input and output, and solution.

Topic	Least Squares Approximations (4.3, Q22)
Original question	Find the best line $C + Dt$ to fit $b = 4, 3, -1, 0, 0$ at times $t = -2, -1, 0, 1, 2$ .
Codex input	Find the best fit line for $x = \text{np.array}([-2, -1, 0, 1, 2])$ and $y = \text{np.array}([4, 3, -1, 0, 0])$
Codex output	<pre>import numpy as np import matplotlib.pyplot as plt  x = np.array([-2, -1, 0, 1, 2]) y = np.array([4, 3, -1, 0, 0])  plt.plot(x, y, 'ro') plt.show()  # y = mx + c # m = (mean(x) * mean(y) - mean(x*y)) / (mean(x)^2 - #     mean(x^2)) # c = mean(y) - m * mean(x)  m = (np.mean(x) * np.mean(y) - np.mean(x*y)) / (np.mean(x)**2     - np.mean(x**2)) c = np.mean(y) - m * np.mean(x)  print(m, c)  y_pred = m * x + c  plt.plot(x, y, 'ro') plt.plot(x, y_pred) plt.show()</pre>
Solution	The best line is $b = 1 - t$ .

MIT 18.06, Question 17: Original question, Codex input and output, and solution.

Topic	Properties of Determinants (5.1, Q1)
Original question	If a 4 by 4 matrix has $\det(A)=1/2$ , find $\det(2A)$ , $\det(-1A)$ , $\det(A^2)$ and $\det(A^{-1})$ .
Codex input	The determinant of a 4 by 4 matrix A is 0.5. What is $\det(2A)$ , $\det(-1A)$ , $\det(A^2)$ , and $\det(A^{-1})$ ? Write a function to calculate the determinant of 2A is 16 times the determinant of A. Write a function to calculate the determinant of -1A is -1 raised to the fourth power times the determinant of A. Write a function to calculate the determinant of $A^2$ is half the determinant of A. Write a function to calculate the determinant of $A^{-1}$ is 1/the determinant of A.
Codex output	<pre>def det2A(A):     return 16*det(A)  def detMinus1A(A):     return -1**4*det(A)  def detA2(A):     return det(A)/2  def detAinv(A):     return 1/det(A)</pre>
Solution	$\det(2A)=2^4*\det(A)=8$ , $\det(-A)=(-1)^4\det(A)=0.5$ , $\det(A^2)=0.25$ , $\det(A^{-1})=2$

MIT 18.06, Question 18: Original question, Codex input and output, and solution.

Topic	Cramer's Rule, Inverses and Volumes (5.3, Q16a)
Original question	Find the area of the parallelogram with edges $v=(3,2)$ and $w=(1,4)$ .
Codex input	Find the area of the parallelogram with edges $v=(3,2)$ and $w=(1,4)$ .
Codex output	<pre># Solution  v = (3, 2) w = (1, 4) area = abs(v[0] * w[1] - v[1] * w[0]) print(area)</pre>
Solution	10

MIT 18.06, Question 19: Original question, Codex input and output, and solution.

Topic	Introduction to Eigenvalues (6.1, Q5)
Original question	Find the eigenvalues of A and B (easy for triangular matrices) and $A + B$ : $A = \begin{bmatrix} 3 & 0 \\ 1 & 1 \end{bmatrix}$ , $B = \begin{bmatrix} 1 & 1 \\ 0 & 3 \end{bmatrix}$ , $A+B = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$ .
Codex input	Find the eigenvalues of A and B (easy for triangular matrices) and $A + B$ : $A = \begin{bmatrix} 3 & 0 \\ 1 & 1 \end{bmatrix}$ , $B = \begin{bmatrix} 1 & 1 \\ 0 & 3 \end{bmatrix}$ , $A+B = \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$ .
Codex output	<pre>import numpy as np  A = np.array([[3,0],[1,1]]) B = np.array([[1,1],[0,3]])  print(np.linalg.eigvals(A)) print(np.linalg.eigvals(B)) print(np.linalg.eigvals(A+B))</pre>
Solution	A and B have eigenvalues 1 and 3

MIT 18.06, Question 20: Original question, Codex input and output, and solution.

Topic	Introduction to Eigenvalues (6.1, Q10)
Original question	Find the eigenvalues and eigenvectors for both of these Markov matrices A and Ainf. A = [.6, .2; .4, .8]. Ainf = [1/3, 1/3; 2/3, 2/3].
Codex input	Find the eigenvalues and eigenvectors for both of these Markov matrices A and Ainf. A = np.array([[.6, .2], [.4, .8]]). Ainf = np.array([[1/3, 1/3], [2/3, 2/3]]).
Codex output	<pre>A = np.array([[.6, .2], [.4, .8]]) A_inf = np.array([[1/3, 1/3], [2/3, 2/3]])  eig_vals, eig_vecs = np.linalg.eig(A) eig_vals_inf, eig_vecs_inf = np.linalg.eig(A_inf)  print(eig_vals) print(eig_vecs) print(eig_vals_inf) print(eig_vecs_inf)</pre>
Solution	A – Eigenvalues: 1 and 0.4, with eigenvectors (1, 2) and x2 = (1, -1). Ainf – Eigenvalues: 1 and (0.4) <sup>100</sup> , same eigenvectors.

MIT 18.06, Question 21: Original question, Codex input and output, and solution.

Topic	Symmetric Matrices (6.4, Q9a)
Original question	Find a symmetric matrix [1,b;1] that has a negative eigenvalue.
Codex input	A = [1,b;1] is a symmetric matrix Find a value of b such that any one eigenvalue of A is negative.
Codex output	<pre>""" A = [[1,b],[b,1]] is a symmetric matrix Find a value of b such that any one eigenvalue of A is negative. """  import numpy as np import scipy.linalg as la  def find_b(n):     """     Find a value of b such that any one eigenvalue of A is     negative.     """     A = np.array([[1,b],[b,1]])     eigvals = la.eigvals(A)     if any(eigvals &lt; 0):         return b     else:         return find_b(n+1)  b = find_b(1) print(b)</pre>
Solution	b=2

MIT 18.06, Question 22: Original question, Codex input and output, and solution.



Topic	Positive Definite Matrices (6.5, Q3)
Original question	For which numbers b is the following matrices positive definite? $S=[1,b;b,9]$ , $S=[2,4;4,c]$ , $S=[c,b;b,c]$ .
Codex input	For which numbers b is the following matrices positive definite? $S=[1,b;b,9]$ , $S=[2,4;4,c]$ , $S=[c,b;b,c]$ . Write a program to search a value x and y for all three matrices to be positive definite: $\text{np.array}([[1,x],[x,9]])$ , $\text{np.array}([[2,4],[4,y]])$ , and $\text{np.array}([[y,x],[x,y]])$ .
Codex output	<pre>import numpy as np  def is_pos_def(x):     return np.all(np.linalg.eigvals(x) &gt; 0)  def main():     b = np.arange(1,10)     c = np.arange(1,10)     for i in b:         for j in c:             S1 = np.array([[1,i],[i,9]])             S2 = np.array([[2,4],[4,j]])             S3 = np.array([[j,i],[i,j]])             if is_pos_def(S1) and is_pos_def(S2) and                 is_pos_def(S3):                 print('S1 = {}, S2 = {}, S3 =                     {}'.format(S1,S2,S3))</pre>
Solution	(a) $-3 < b < 3$ , (b) $c > 8$ , (c) $c > b$

MIT 18.06, Question 23: Original question, Codex input and output, and solution.

Topic	Image processing by Linear Algebra (7.1, Q6)
Original question	Find the eigenvalues and the singular values of this 2 by 2 matrix $A=[2,1;4,2]$ . The eigenvectors (1,2) and (1,-2) of A are not orthogonal.
Codex input	Write a function to find the eigenvalues and the singular values of this 2 by 2 matrix $A=[2,1;4,2]$ . The eigenvectors (1,2) and (1,-2) of A are not orthogonal.
Codex output	<pre>import numpy as np  A = np.array([[2,1],[4,2]])  eigenvalues, eigenvectors = np.linalg.eig(A)  print(eigenvalues) print(eigenvectors)  singular_values = np.linalg.svd(A)  print(singular_values)</pre>
Solution	Eigenvalues are 4 and 0

MIT 18.06, Question 24: Original question, Codex input and output, and solution.

Topic	Principal Component Analysis (7.3,Q1)
Original question	Suppose A0 holds these 2 measurements of 5 samples: $A_0 = [5,4,3,2,1;-1,1,0,1,-1]$ . Find the average of each row and subtract it to produce the centered matrix A. Compute the sample covariance matrix $S=AA'/(n-1)$ and find its eigenvalues. What line through the origin is closest to the 5 samples in the columns of A?
Codex input	Suppose A0 holds these 2 measurements of 5 samples: $A_0 = [5,4,3,2,1;-1,1,0,1,-1]$ . Find the average of each row and subtract it to produce the centered matrix A. Compute the sample covariance matrix $S=AA'/(n-1)$ and find its eigenvalues. What line through the origin is closest to the 5 samples in the columns of A?
Codex output	<pre> ##### Suppose A0 holds these 2 measurements of 5 samples: A0 = [5,4,3,2,1;-1,1,0,1,-1]. Find the average of each row and subtract it to produce the centered matrix A. Compute the sample covariance matrix S=AA^T/(n-1) and find its eigenvalues. #####  import numpy as np  A0 = np.array([[5,4,3,2,1],[-1,1,0,1,-1]]) A = A0 - np.mean(A0, axis=1).reshape(2,1) S = np.dot(A, A.T) / (5-1) eigvals, eigvecs = np.linalg.eig(S)  print(eigvals) print(eigvecs)  ##### What line through the origin is closest to the 5 samples in the columns of A? #####  import numpy as np  A0 = np.array([[5,4,3,2,1],[-1,1,0,1,-1]]) A = A0 - np.mean(A0, axis=1).reshape(2,1) S = np.dot(A, A.T) / (5-1) eigvals, eigvecs = np.linalg.eig(S)  print(eigvecs[:,0]) </pre>
Solution	The vertical line passing through the origin is closest to the five datapoints.

MIT 18.06, Question 25: Original question, Codex input and output, and solution.

Topic	The Idea of a Linear Transformation (8.1, Q12)
Original question	Suppose a linear T transforms (1,1) to (2,2) and (2,0) to (0,0). Find T(v), when $v=[2,2]$ , $v=(3,1)$ , $v=(-1,1)$ , $v=(a,b)$ .
Codex input	Use a while True loop to find a random matrix M such that $M \cdot \text{np.array}([[1,1],[2,2]]) = \text{np.array}([[2,2],[0,0]])$ . Find $M \cdot \text{np.array}([2,2])$ . Find $M \cdot \text{np.array}([3,1])$ . Find $M \cdot \text{np.array}([-1,1])$ . Use Sympy to find $M \cdot \text{np.array}([a,b])$ .
Codex output	<pre>import numpy as np import sympy as sp  while True:     M = np.random.randint(1,10,(2,2))     if np.array_equal(M.dot(np.array([[1,1],[2,2]])), np.array([[2,2],[0,0]])):         print(M)         print(M.dot(np.array([2,2])))         print(M.dot(np.array([3,1])))         print(M.dot(np.array([-1,1])))         a,b = sp.symbols('a b')         print(M.dot(np.array([a,b])))         break</pre>
Solution	$T(v) = (4, 4); (2, 2); (2, 2)$ ; if $v = (a, b) = b(1, 1) + a-b 2 (2, 0)$ then $T(v) = b(2, 2) + (0, 0)$

MIT 18.06, Question 26: Original question, Codex input and output, and solution.

Topic	Linear Algebra in Probability and Statistics (12.1, Q2)
Original question	We know: $1/3$ of all integers are divisible by 3 and $1/7$ of integers are divisible by 7. What fraction of integers will be divisible by 3 or 7 or both ?
Codex input	We know: $1/3$ of all integers are divisible by 3 and $1/7$ of integers are divisible by 7. Write a program to calculate the fraction of integers will be divisible by 3 or 7 or both ?
Codex output	<pre># Solution:  # import fractions module import fractions  # define a function to calculate the fraction of integers divisible by 3 or 7 or both def divisible_by_3_or_7():     # calculate the fraction of integers divisible by 3 or 7     or both     return fractions.Fraction(1, 3) + fractions.Fraction(1, 7) - fractions.Fraction(1, 21)  # print the result print(divisible_by_3_or_7())</pre>
Solution	If we add $1/3$ to $1/7$ (fraction of integers divisible by 3 plus fraction divisible by 7) we have double counted the integers divisible by both 3 and 7. This is a fraction $1/21$ of all integers (because these double counted numbers are multiples of 21). So the fraction divisible by 3 or 7 or both is $1/3+1/7-1/21=7/21+3/21-1/21=9/21=3/7$

MIT 18.06, Question 27: Original question, Codex input and output, and solution.

Topic	Linear Algebra in Probability and Statistics (12.1, Q4)
Original question	Sample again from 1 to 1000 but look at the last digit of the sample squared. That square could end with $x = 0, 1, 4, 5, 6,$ or $9$ . What are the probabilities $p_0, p_1, p_4, p_5, p_6, p_9$ ?
Codex input	Suppose you sample from the numbers 1 to 1000 with equal probabilities $1/1000$ and then square the number. What are the probabilities $p_0$ to $p_9$ that the last digit of your sample is $0, \dots, 9$ ?
Codex output	<pre>def prob_last_digit(n):     """     n: number of samples     """     # generate n samples     samples = np.random.randint(1, 1000, size=n)     # square the samples     samples = np.power(samples, 2)     # get the last digit of each sample     samples = samples % 10     # count the number of times each digit appears     counts = np.bincount(samples)     # normalize the counts     counts = counts / n     return counts  # print the probabilities print(prob_last_digit(1000000))</pre>
Solution	For numbers ending in $0, 1, 2, \dots, 9$ the squares end in $x = 0, 1, 4, 9, 6, 5, 6, 9, 4, 1$ . So the probabilities of $x = 0$ and $5$ are $p = 1/10$ and the probabilities of $x = 1, 4, 6, 9$ are $p = 1/5$ The mean is $m = \sum p_i x_i = 0/0 + 5/10 + (1/5)*(1 + 4 + 6 + 9) = 4.5 =$ same as before

MIT 18.06, Question 28: Original question, Codex input and output, and solution.

Topic	Linear Algebra in Probability and Statistics (12.1, Q8)
Original question	If all 24 samples from a population produce the same age $x = 20$ , what are the sample mean $\mu$ and the sample variance $S^2$ ?
Codex input	If all 24 samples from a population produce the same age $x = 20$ , what are the sample mean $\mu$ and the sample variance $S^2$ ?
Codex output	<pre># mu = 20 # S^2 = 0</pre>
Solution	$\mu = 20, S^2 = 0$

MIT 18.06, Question 29: Original question, Codex input and output, and solution.

Topic	Linear Algebra in Probability and Statistics (12.1, Q9)
Original question	Find the average $A_N$ of a million random 0-1 samples! What is $X = (A_N - 1/2)/(2*\sqrt{N})$ ?
Codex input	Find the average $A_N$ of a million random 0-1 samples! What is $X = (A_N - 1/2)/(2*\sqrt{N})$ ?
Codex output	<pre> import numpy as np import matplotlib.pyplot as plt  def get_average(N):     """     N: number of samples     """     samples = np.random.randint(0, 2, N)     return np.mean(samples)  def get_X(N):     """     N: number of samples     """     return (get_average(N) - 0.5) / (2*np.sqrt(N))  def get_X_list(N):     """     N: number of samples     """     X_list = []     for i in range(N):         X_list.append(get_X(N))     return X_list  def get_X_mean(N):     """     N: number of samples     """     return np.mean(get_X_list(N))  def get_X_std(N):     """     N: number of samples     """     return np.std(get_X_list(N))  N = 1000000 print(get_X_mean(N)) print(get_X_std(N))  plt.hist(get_X_list(N), bins=100) plt.show() </pre>
Solution	$X = (A_N - 1/2)/(2\sqrt{N}) = (A_N - 1/2)/2000$ for $N =$ one million.

MIT 18.06, Question 30: Original question, Codex input and output, and solution.

Topic	Diophantine Equations, Divisibility, GCD, Euclidean Algorithm, Primes, Binomial Coefficients, Congruences
Original Question	Use the Euclidean algorithm to find an integer solution $(x_0, y_0)$ to $89x + 43y = 1$ .
Codex Input	In number theory, write a program that uses the Euclidean algorithm to find an integer solution $(x_0, y_0)$ to $89x + 43y = 1$ .
Codex Output	<pre>def gcd(a, b):     if b == 0:         return a     else:         return gcd(b, a % b)  def extended_gcd(a, b):     if b == 0:         return (1, 0, a)     else:         (x, y, d) = extended_gcd(b, a % b)         return (y, x - (a // b) * y, d)  def solve(a, b, c):     d = gcd(a, b)     if c % d != 0:         return None     else:         (x, y, d) = extended_gcd(a, b)         return (x * (c // d), y * (c // d))  print(solve(89, 43, 1))</pre>
Solution	$(x_0, y_0) = (-14, 29)$

Question 1 from MIT 18.781

Topic	Fermat, Euler, Wilson, Linear Congruences
Original Question	Let $p$ be an odd prime. How many $p$ -element subsets of $1, 2, \dots, 2p$ have the sum of their elements divisible by $p$ ?
Codex Input	Implement <code>n choose r</code> . Then, write a program to find how many $p$ -element subsets of $1, 2, \dots, 2p$ have the sum of their elements divisible by $p$ . Let $p$ be an odd prime. You can do this by computing $2p + (2p \text{ choose } p) - 2$ , all divided by $p$ .
Codex Output	<pre>def ncr(n, r):     if r == 0:         return 1     if r == 1:         return n     if r == n:         return 1     return ncr(n-1, r-1) + ncr(n-1, r)  def main():     p = int(input())     print((2*p + ncr(2*p, p) - 2) // p)  if __name__ == "__main__":     main()</pre>
Solution	$(2p + (2pCp - 2))/p$

Question 2 from MIT 18.781

Topic	Linear Congruences, Chinese Remainder Theorem, Algorithms, Primality, Factoring, RSA, Hensel's Lemma
Original Question	What are the last two digits of $2^{100}$ and of $3^{100}$ ?
Codex Input	Write a program to print the last two digits of $2^{100}$ and of $3^{100}$ .
Codex Output	<pre>print (2**100%100) print (3**100%100)</pre>
Solution	76 and 01

Question 3 from MIT 18.781

Topic	Linear Congruences, Chinese Remainder Theorem, Algorithms, Primality, Factoring, RSA, Hensel's Lemma
Original Question	Find the number of solutions of $x^2 \equiv x \pmod{m}$ for any positive integer $m$
Codex Input	Write a program to count the solutions to $x**2 \% m == x$ where $m$ is positive. $x$ is an integer and $x \geq 0$ and $x < m$ .
Codex Output	<pre>def count_solutions(m):     count = 0     for x in range(m):         if x**2 % m == x:             count += 1     return count</pre>
Solution	For an arbitrary integer $m$ , the number of solutions is $2^r$ where $r$ is the number of distinct prime divisors of $m$ .

Question 4 from MIT 18.781

Topic	Linear Congruences, Chinese Remainder Theorem, Algorithms, Primality, Factoring, RSA, Hensel's Lemma
Original Question	Show that the number $n = 561 = 3 * 11 * 17$ satisfies the property P: for any $a$ coprime to $n$ , we have $a^{(n-1)} \equiv 1 \pmod{n}$
Codex Input	Write a program to show all numbers coprime to 561, $x**(560) \% 561 == 1$ .
Codex Output	<pre>def coprime(n):     for i in range(2, n):         if n % i == 0:             return False     return True  for i in range(2, 561):     if i**(560) % 561 == 1:         print(i)</pre>
Solution	We need to show that $a^{560} \equiv 1 \pmod{3}$ , $\pmod{11}$ , and $\pmod{17}$ for any $a$ coprime to 561. Since $a$ is coprime to 3, $a^2 \equiv 1 \pmod{3}$ , so $a^{560} = a^{(2*280)} \equiv 1 \pmod{3}$ . Since $a$ is coprime to 11, $a^{10} \equiv 1 \pmod{11}$ , so $a^{560} = a^{(56*10)} \equiv 1 \pmod{11}$ . Since $a$ is coprime to 17, $a^{16} \equiv 1 \pmod{17}$ , so $a^{560} = a^{(35*16)} \equiv 1 \pmod{17}$ .

Question 5 from MIT 18.781

Topic	Linear Congruences, Chinese Remainder Theorem, Algorithms, Primality, Factoring, RSA, Hensel's Lemma
Original Question	This computational exercise will involve the notion of "density". We say that a set $S$ of primes has density $d$ if the limit of $\frac{ S \cap \{p < N : p \text{ prime}\} }{ \{p < N : p \text{ prime}\} }$ as $N$ goes to infinity exists and is equal to $d$ . Let $f(x) = x^3 - 2$ , write a program to calculate the set $S$ of primes $p$ less than 10000 such that $f$ has a solution modulo $p$ . Approximate the density of such primes.
Codex Input	Find the number of primes $< 10000$ . Find the fraction of primes $< 10000$ that there is some integer between 0 and $p$ such that $(x^3 - 2) \% p == 0$ . Make sure to divide by total number of primes $< 10000$ .
Codex Output	<pre> from math import sqrt  def is_prime(n):     if n == 2:         return True     if n % 2 == 0 or n &lt; 2:         return False     for i in range(3, int(sqrt(n)) + 1, 2):         if n % i == 0:             return False     return True  def main():     primes = [2]     for i in range(3, 10000, 2):         if is_prime(i):             primes.append(i)     print(len(primes))     count = 0     for p in primes:         for x in range(1, p):             if (x**3 - 2) % p == 0:                 count += 1                 break     print(count / len(primes))  if __name__ == '__main__':     main() </pre>
Solution	$\approx 2/3$

Question 6 from MIT 18.781



Topic	Congruences mod Primes, Order, Primitive Roots, Primitive Roots (Prime Powers), Index Calculus, Quadratic Residues, Quadratic Reciprocity
Original Question	Find a primitive root modulo 23 and modulo $23^3$ .
Codex Input	In Number Theory, write a program to compute a primitive root of modulo 23. In Number Theory, write a program to compute a primitive root of modulo $23*23*23$ .
Codex Output	<pre> def primitive_root(modulo):     """     Returns a primitive root of modulo.     """     if modulo == 2:         return 1     roots = []     required_set = set(num for num in range(1, modulo) if         gcd(num, modulo) == 1)      for g in range(1, modulo):         actual_set = set(pow(g, powers) % modulo for powers             in range(1, modulo))         if required_set == actual_set:             roots.append(g)     return roots  if __name__ == '__main__':     print(primitive_root(23))  def primitive_root(modulo):     required_set = set(num for num in range (1, modulo) if         gcd(num, modulo) == 1)     for g in range(1, modulo):         actual_set = set(pow(g, powers) % modulo for powers             in range (1, modulo))         if required_set == actual_set:             return g     return 0  if __name__ == '__main__':     print(primitive_root(23*23*23)) </pre>
Solution	mod 23 is [5, 7, 10, 11, 14, 15, 17, 19, 20, 21], mod $23^3$ is [5, 7, 10, 11, 14, 15, 17, 19, 20, 21, 30, 33, 34, 37, 38, 40, 43, 44, 51, 53, 56, 57, 60, 61, 65, 66, 67, 74, 76, 79, 80, 83, 84, 86, 88, 89, 90, 97, 99, 102, 103, 106, 107, 109, 111, 112, 113, 120, 122, 125, 126, 129, 132, 134, 135, 136, 143, 145, 148, 149, 152, 153, 155, 157, 158, 159, 166, 168, 171, 172, 175, 176, 178, 180, 181, 182, 189, 191, 194, 198, 199, 201, 203, 204, 205, 212, 214, 217, 218, 221, 222, 224, 226, 227, 228, 235, 237, 240, 241, 244, 245, 247, 249, 250, 251, 258, 260, 264, 267, 268, 270, 272, 273, 281, 283, 286, 287, 290, 291, 293, 295, 296, 297, 304, 306, 309, 310, 313, 314, 316, 318, 319, 320, 327, 329, 332, 333, 336, 337, 339, 341, 342, 343, 350, 355, 356, 360, 362, 364, 365, 366, 373, 375, 378, 379, 382, 383, 385, 387, 388, 389, 396, 398, 401, 402, 405, 406, 408, 410, 412, 419, 421, 424, 425, 428, 429, 431, 433, 434, 435, 442, 444, 447, 448, 451, 452, 454, 456, 457, 458, 465, 467, 470, 471, 474, 475, 477, 479, 480, 481, 488, 490, 493, 494, 497, 498, 500, 502, 503, 504, 511, 513, 516, 517, 520, 521, 523, 525, 526, 527]

Question 7 from MIT 18.781

Topic	Tonelli's Algorithm
Original Question	Solve $x^2 \equiv 21 \pmod{41}$ using Tonelli's algorithm.
Codex Input	Write a program to solve $x^2 \equiv 21 \pmod{41}$ using Tonelli's algorithm.
Codex Output	<pre>def tonelli(n, p):     assert n &lt; p     assert legendre(n, p) == 1, "not a square (mod p)"     q = p - 1     s = 0     while q % 2 == 0:         q //= 2         s += 1     if s == 1:         return pow(n, (p + 1) // 4, p)     for z in range(2, p):         if p - 1 == legendre(z, p):             break     c = pow(z, q, p)     r = pow(n, (q + 1) // 2, p)     t = pow(n, q, p)     m = s     t2 = 0     while (t - 1) % p != 0:         t2 = (t * t) % p         for i in range(1, m):             if (t2 - 1) % p == 0:                 break         t2 = (t2 * t2) % p     b = pow(c, 1 &lt;&lt; (m - i - 1), p)     r = (r * b) % p     c = (b * b) % p     t = (t * c) % p     m = i     return r  def legendre(a, p):     return pow(a, (p - 1) // 2, p)  print(tonelli(21, 41))</pre>
Solution	21

Question 8 from MIT 18.781

Topic	Primes Congruent to 1 mod n
Original Question	Does the following congruence $x^2 \equiv -2 \pmod{118}$ have solutions, if so how many?
Codex Input	Use sympy. Solve for x by setting $x^2 \% 118 = -2 \% 118$ . Use Eq(a, b) for saying a and b are equal.
Codex Output	<pre>from sympy import *  x = Symbol('x')  solve(Eq(x**2 % 118, -2 % 118), x)</pre>
Solution	There are two solutions to the congruence.

Question 9 from MIT 18.781

Topic	Primes Congruent to 1 mod n
Original Question	Does the following congruence $x^2 \equiv -1 \pmod{244}$ have solutions, if so how many?
Codex Input	Use sympy. Solve for all answers of x by setting $\text{Eq}(x^2\%244, -1\%244)$ .
Codex Output	<pre> from sympy import * x = Symbol('x') solve(Eq(x**2 % 244, -1 % 244)) </pre>
Solution	None

Question 10 from MIT 18.781

Topic	Primes Congruent to 1 mod n
Original Question	Does the following congruence $x^2 \equiv -1 \pmod{365}$ have solutions, if so how many?
Codex Input	Use sympy. Solve for all answers of x by setting $\text{Eq}(x^2\%365, -1\%365)$
Codex Output	<pre> from sympy import * x = Symbol('x') solve(Eq(x**2 % 365, -1 % 365)) </pre>
Solution	There are four solutions to the congruence.

Question 11 from MIT 18.781

Topic	Primes Congruent to 1 mod n
Original Question	Does the following congruence $x^2 \equiv 267 \pmod{789}$ have solutions, if so how many?
Codex Input	Use sympy. Solve for x by setting $x^2\%789 = 267\%789$ . Use Eq(a, b) for saying a and b are equal. Use Eq(a, b) for saying a and b are equal.
Codex Output	<pre> from sympy import * x = Symbol('x') eq = Eq(x**2 % 789, 267 % 789) print(solve(eq, x)) </pre>
Solution	There are two solutions to the congruence.

Question 12 from MIT 18.781

Topic	Generating Functions
Original Question	At a party, there are $n$ people, who each give their coat to a coat-check person. Calculate the probability the coats can be handed back, each person receiving one, so that no one receives their own coat.
Codex Input	A party has $n$ people who each give their coat. Write a program to calculate the probability that the coats can be handed back, each person receiving one, so that no one receives their own coat. Begin by creating a list in which the value of each element is its index. Then, shuffle the list and check if any of the values match its index. Return the percentage of lists for which the check was False.
Codex Output	<pre> import random  def coat_check(n):     people = list(range(n))     random.shuffle(people)     for i in range(n):         if people[i] == i:             return False     return True  def coat_check_sim(n, sims):     count = 0     for i in range(sims):         if coat_check(n):             count += 1     return count/sims  print(coat_check_sim(10, 10000)) </pre>
Solution	The probability that no one receives their coat back is approximately $1/e$ or about 36.79%.

Question 13 from MIT 18.781

Topic	Continued Fractions
Original Question	Calculate the continued fractions of 6157/783.
Codex Input	Write a program that calculates the continued fractions of 6157/783.
Codex Output	<pre>def continued_fraction(n, d):     """     Calculate the continued fraction of n/d.     """     a = []     while d != 0:         a.append(n//d)         n, d = d, n % d     return a  def convergents(a):     """     Calculate the convergents of a continued fraction.     """     h = [0, 1]     k = [1, 0]     for i in range(len(a)):         h.append(a[i]*h[i+1] + h[i])         k.append(a[i]*k[i+1] + k[i])     return h[2:], k[2:]  def main():     a = continued_fraction(6157, 783)     print(a)     h, k = convergents(a)     print(h[-1], k[-1])  if __name__ == '__main__':     main()</pre>
Solution	[7, 1, 6, 3, 6, 1, 4]

Question 14 from MIT 18.781

Topic	Continued Fractions
Original Question	Calculate the continued fractions of sqrt(15).
Codex Input	Write a program that calculates the continued fractions of sqrt(15).
Codex Output	<pre>def continued_fraction(n):     """     Calculate the continued fraction of sqrt(n)     """     m = 0     d = 1     a = a0 = int(n**0.5)     result = [a0]     while a != 2*a0:         m = d*a - m         d = (n - m**2)/d         a = int((a0 + m)/d)         result.append(a)     return result  print(continued_fraction(15))</pre>
Solution	[3, 1, 6, 1, ...]

Question 15 from MIT 18.781

Topic	GCD
Original Question	Find the gcd of 621 and 483.
Codex Input	Find the gcd of 621 and 483.
Codex Output	<pre>def gcd(a, b):     if b == 0:         return a     else:         return gcd(b, a % b)  print(gcd(621, 483))</pre>
Solution	69

Question 16 from MIT 18.781

Topic	Exponential modulus
Original Question	Find $3^{64} \bmod 67$
Codex Input	Find $3^{64} \bmod 67$
Codex Output	<pre>def main():     """     Find 3^64 mod 67.     """     print(pow(3, 64, 67))  if __name__ == "__main__":     main()</pre>
Solution	15

Question 17 from MIT 18.781

Topic	Congruences
Original Question	How many solutions are there to $x^{12} \equiv 7 \pmod{19}$ ?
Codex Input	Write a program to compute the number of solutions to $x^{12} \equiv 7 \pmod{19}$
Codex Output	<pre>def main():     print(solve(19, 7))  def solve(p, x):     count = 0     for i in range(p):         if (i**12) % p == x:             count += 1     return count  if __name__ == "__main__":     main()</pre>
Solution	6

Question 18 from MIT 18.781

Topic	Divisors
Original Question	For $n = 1728$ , figure out the number of positive divisors of $n$ , and the sum of its positive divisors.
Codex Input	Let $n = 1728$ . Write a function to compute the number of positive divisors of $n$ and the sum of the positive divisors.
Codex Output	<pre> def divisors(n):     divisors = []     for i in range(1, n+1):         if n % i == 0:             divisors.append(i)     return divisors  def sum_divisors(n):     divisor = divisors(n)     sum_divisors = 0     for i in divisor:         sum_divisors += i     return sum_divisors  def main():     n = 1728     print(divisors(n))     print(sum_divisors(n))  if __name__ == "__main__":     main() </pre>
Solution	5080

Question 19 from MIT 18.781

Topic	Linear Congruences, Chinese Remainder Theorem, Algorithms, Primality, Factoring, RSA,Hensel's Lemma
Original Question	Tabulate the number of primes less than x, for x = 10000, 20000,...., 100000. Also tabulate the number of primes less than x and of the form 4k +1, and the number of the form 4k + 3
Codex Input	Write a program to tabulate the number of primes less than x, for x = 10000, 20000,...., 100000. Also tabulate the number of primes less than x and of the form 4k +1, and the number of the form 4k + 3
Codex Output	<pre> def is_prime(n):     if n == 1:         return False     elif n == 2:         return True     elif n % 2 == 0:         return False     else:         for i in range(3, int(n**0.5)+1, 2):             if n % i == 0:                 return False         return True  def is_prime_4k_1(n):     if n == 1:         return False     elif n == 2:         return True     elif n % 2 == 0:         return False     else:         for i in range(3, int(n**0.5)+1, 2):             if n % i == 0:                 return False         if n % 4 == 1:             return True         else:             return False  def is_prime_4k_3(n):     if n == 1:         return False     elif n == 2:         return True     elif n % 2 == 0:         return False     else:         for i in range(3, int(n**0.5)+1, 2):             if n % i == 0:                 return False         if n % 4 == 3:             return True         else:             return False  def main():     for i in range(10000, 100001, 10000):         count = 0         for j in range(i, i+10000):             if is_prime(j):                 count += 1         print(i, count)     for i in range(10000, 100001, 10000):         count = 0         for j in range(i, i+10000):             if is_prime_4k_1(j):                 count += 1         print(i, count)     for i in range(10000, 100001, 10000):         count = 0         for j in range(i, i+10000):             if is_prime_4k_3(j):                 count += 1         print(i, count)  if __name__ == '__main__':     main() </pre>
Solution	You should notice that about 50 percent of the primes are 1 mod 4 and about 50 percent are 3



**G Harvard STAT110: Probability**

Topic	Complementary Probability
Original Question	In any 15-minute interval, there is a 20% probability that you will see at least one shooting star. What is the probability that you see at least one shooting star in the period of an hour?
Codex Input	In any 15-minute interval, there is a 20% probability that you will see at least one shooting star.  There are 4 15-minute intervals in an hour.  Write a simulation program to estimate the probability that you see at least one shooting star in the period of an hour.
Codex Output	<pre>import random  def star():     return random.random() &lt;= 0.2  def star_in_hour():     count = 0     for i in range(4):         if star():             count += 1     return count &gt;= 1  def estimate_probability():     count = 0     for i in range(10000):         if star_in_hour():             count += 1     return count / 10000  print(estimate_probability())</pre>
Solution	0.594

Question 1 from STAT110

Topic	Conditional Probability
Original Question	A certain couple tells you that they have two children, at least one of which is a girl. What is the probability that they have two girls?
Codex Input	A couple has two children, with at least one child being a girl. Write a simulation program to estimate the probability that the both children are girls.
Codex Output	<pre>import random  def random_kid():     return random.choice(["boy", "girl"])  both_girls = 0 older_girl = 0 either_girl = 0  random.seed(0) for _ in range(10000):     younger = random_kid()     older = random_kid()     if older == "girl":         older_girl += 1     if older == "girl" and younger == "girl":         both_girls += 1     if older == "girl" or younger == "girl":         either_girl += 1  print("P(both   older):", both_girls / older_girl) print("P(both   either):", both_girls / either_girl)</pre>
Solution	1/3

Question 2 from STAT110

Topic	Conditional Probability
Original Question	You have a group of couples that decide to have children until they have their first girl, after which they stop having children. What is the expected gender ratio of the children that are born? What is the expected number of children each couple will have?
Codex Input	N couples decide to have children (both boys and girls) until they have their first girl, after which they stop having children.  Write a sampling program to estimate the ratio of boys to girls after all N couples stop having children.
Codex Output	<pre> import random  def children():     return random.choice(['boy', 'girl'])  def children_until_girl(n):     boys = 0     girls = 0     for i in range(n):         child = children()         if child == 'boy':             boys += 1         else:             girls += 1             break     return (boys, girls)  def children_until_girl_sampling(n, samples):     boys = 0     girls = 0     for i in range(samples):         child = children_until_girl(n)         boys += child[0]         girls += child[1]     return (boys, girls)  def children_until_girl_ratio(n, samples):     boys, girls = children_until_girl_sampling(n, samples)     return float(boys) / float(girls)  print(children_until_girl_ratio(5, 10000)) </pre>
Solution	1/2

Question 3 from STAT110

Topic	Expected Value
Original Question	A lazy high school senior types up application and envelopes to n different colleges, but puts the applications randomly into the envelopes. What is the expected number of applications that went to the right college?
Codex Input	A lazy high school senior types up application and envelopes to n different colleges, but puts the applications randomly into the envelopes. Write a program to calculate the expected number of applications that were put in the right envelopes.
Codex Output	<pre>import random  def main():     n = int(input("Enter the number of colleges: "))     num_trials = int(input("Enter the number of trials: "))     count = 0     for i in range(num_trials):         applications = [i for i in range(n)]         envelopes = [i for i in range(n)]         random.shuffle(envelopes)         for i in range(n):             if applications[i] == envelopes[i]:                 count += 1     print("The expected number of applications that were     put in the right envelopes is", count/num_trials)  if __name__ == "__main__":     main()</pre>
Solution	1

#### Question 4 from STAT110

Topic	Expected Value and Conditional Expectation
Original Question	What's the expected number of coin flips until you get two heads in a row? What's the expected number of coin flips until you get two tails in a row?
Codex Input	Generate 1000 sequences of random coin flips, and end each sequence once you get two consecutive heads. Write a program that computes the average sequence length.
Codex Output	<pre>import random  def flip_coin():     return random.choice(['H', 'T'])  def flip_coin_until_consecutive_heads():     seq = ''     while True:         seq += flip_coin()         if seq[-2:] == 'HH':             break     return seq  def flip_coin_until_consecutive_heads_1000_times():     seqs = []     for i in range(1000):         seqs.append(flip_coin_until_consecutive_heads())     return seqs  def compute_average_sequence_length():     seqs = flip_coin_until_consecutive_heads_1000_times()     return sum([len(seq) for seq in seqs]) / len(seqs)  print(compute_average_sequence_length())</pre>
Solution	6

#### Question 5 from STAT110

Topic	Expected Value
Original Question	Let's say we play a game where I keep flipping a coin until I get heads. If the first time I get heads is on the nth coin, then I pay you $2^{n-1}$ dollars. How much would you pay me to play this game?
Codex Input	Let's say we play a game where I keep flipping a coin until I get heads.  If the first time I get heads is on the nth coin, then I pay you $2^{n-1}$ dollars.  Write a large-scale simulation program to determine how much you would pay me to play this game.
Codex Output	<pre>import random  def flip_coin():     return random.randint(0,1)  def play_game(n):     heads = False     count = 0     while not heads:         count += 1         heads = flip_coin()     return 2*count - 1  def run_simulation(n):     total = 0     for i in range(n):         total += play_game(n)     return total/n  print(run_simulation(100000))</pre>
Solution	3

Question 6 from STAT110

Topic	Bayes' Theorem and Conditional Probability
Original Question	You have two coins, one of which is fair and comes up heads with a probability $1/2$ , and the other which is biased and comes up heads with probability $3/4$ . You randomly pick coin and flip it twice, and get heads both times. What is the probability that you picked the fair coin?
Codex Input	You have one coin that is fair and comes up heads with probability $1/2$ .  You have another coin that is biased and comes up heads with probability $3/4$ .  You pick either the fair or biased coin with 50-50 probability. You flip the chosen coin twice.  Write a program using Bayes' Theorem to compute and print the probability of having picked the fair coin, given the fact that the chosen coin lands heads both times.
Codex Output	<pre># P(fair H,H) = P(H,H fair) * P(fair) / P(H,H) # P(H,H fair) = P(H fair) * P(H fair) # P(H,H) = P(H fair) * P(H fair) * P(fair) + P(H biased) * # P(H biased) * P(biased) # P(fair H,H) = (1/2 * 1/2 * 1/2) / (1/2 * 1/2 * 1/2 + 3/4 # * 3/4 * 1/2)  print((1/2 * 1/2 * 1/2) / (1/2 * 1/2 * 1/2 + 3/4 * 3/4 * 1/2))</pre>
Solution	4/13

Question 7 from STAT110

Topic	Conditional Probability
Original Question	How many people do we need in a class to make the probability that two people have the same birthday more than 1/2? For simplicity, assume there are 365 days in a year.
Codex Input	Write a program to compute how many people do we need in a class to make the probability that two people have the same birthday more than 1/2? There are 365 days in a year.
Codex Output	<pre>def birthday_paradox(n):     """     n: number of people in the class     """     return 1 - (1 - 1/365)**(n*(n-1)/2)  print(birthday_paradox(23))</pre>
Solution	23

Question 8 from STAT110

Topic	Conditional Probability
Original Question	A bag contains one marble which is either green or blue, with equal probabilities. A green marble is put in the bag (so there are 2 marbles now), and then a random marble is taken out. The marble taken out is green. What is the probability that the remaining marble is also green?
Codex Input	<p>A woman is pregnant with twin boys.  Twins may be either identical or fraternal (non-identical).  In general, only 1/3 of all twins born are identical.  Identical twins must be of the same sex, while fraternal twins may or may not be.  Assume that identical twins are equally likely to be both boys or both girls, while for fraternal twins all possibilities are equally likely.  Given the above information, write a simulation program to estimate the average probability that the woman's twins are identical.</p>
Codex Output	<pre>import random  def green_marble_probability(num_trials):     """     num_trials: number of trials     """     num_green = 0     for i in range(num_trials):         bag = ['green', 'blue']         random.shuffle(bag)         bag.append('green')         random.shuffle(bag)         if bag[0] == 'green':             num_green += 1     return num_green / num_trials  print(green_marble_probability(100000))</pre>
Solution	2/3

Question 9 from STAT110

Topic	Conditional Probability
Original Question	Calvin and Hobbes play a match consisting of a series of games, where Calvin has probability 0.5 of winning each game (independently). They play with a 'win by two' rule: the first player to win two games more than his opponent wins the match. Find the probability that Calvin wins the match (in terms of p)
Codex Input	<p>Calvin and Hobbes play a match consisting of a series of games, where Calvin has probability 0.5 of winning each game (independently).</p> <p>They play with a 'win by two' rule: the first player to win two games more than his opponent wins the match.</p> <p>Write a simulation program to estimate the average probability that Calvin wins the match.</p>
Codex Output	<pre>import random  def calvin_wins(n):     wins = 0     for i in range(n):         if random.random() &lt; 0.5:             wins += 1     return wins &gt;= 2  def calvin_wins_match(n):     wins = 0     for i in range(n):         if calvin_wins(2):             wins += 1     return wins / n  print(calvin_wins_match(100000))</pre>
Solution	0.25

Question 10 from STAT110

Topic	Geometric Probability
Original Question	Two witches make a nightly visit to an all-night coffee shop. Each arrives at a random time between 0:00 and 1:00. Each one of them stays for exactly 30 minutes. On any one given night, what is the probability that the witches will meet at the coffee shop?
Codex Input	Two witches make a nightly visit to an all-night coffee shop. Each arrives at a random time between 0:00 and 1:00. Each witch stays at the coffee shop for exactly 30 minutes. Write a simulation program to estimate the average probability that the witches will see each other at the coffee shop.
Codex Output	<pre>import random  def witches():     witches = []     for i in range(2):         witches.append(random.randint(0, 60))     return witches  def main():     witches_seen = 0     for i in range(100000):         w = witches()         if w[0] &gt; w[1]:             if w[0] - w[1] &lt; 30:                 witches_seen += 1         else:             if w[1] - w[0] &lt; 30:                 witches_seen += 1     print(witches_seen/100000)  main()</pre>
Solution	0.75

Question 11 from STAT110

Topic	Geometric Probability
Original Question	p and q are two points chosen at random between 0 & 1. What is the probability that the ratio p/q lies between 1 & 2?
Codex Input	Roll a die. If the die is 4, 5, or 6, you get paid the value of the roll. If the die is 1, 2, or 3, you reroll and get paid the value of the new roll. Write a program that computes the expected value.
Codex Output	<pre>import random  def prob_p_q():     p = random.random()     q = random.random()     if p/q &gt;= 1 and p/q &lt;= 2:         return True     else:         return False  def prob_p_q_sim(n):     count = 0     for i in range(n):         if prob_p_q():             count += 1     return count/n  print(prob_p_q_sim(100000))</pre>
Solution	1/4

Question 12 from STAT110



Topic	Conditional Expectation
Original Question	Roll a die, and you get paid what the dice shows. But if you want, you can request a second chance & roll the die again; get paid what the second roll shows instead of the first. What is the expected value?
Codex Input	Roll a die, and you get paid what the number on the dice. You can request a second chance and roll the die again. You get paid the number on second roll instead of the first. Write a function to compute the expected value in Monte Carlo.
Codex Output	<pre>import random  def roll_die():     return random.randint(1, 6)  def second_chance():     return random.randint(1, 6)  def monte_carlo(n):     total = 0     for i in range(n):         first_roll = roll_die()         second_roll = second_chance()         total += max(first_roll, second_roll)     return total/n  print(monte_carlo(100000))</pre>
Solution	4.25

Question 13 from STAT110

Topic	Conditional Expectation
Original Question	A very innocent monkey throws a fair die. The monkey will eat as many bananas as are shown on the die, from 1 to 5. But if the die shows '6', the monkey will eat 5 bananas and throw the die again. This may continue indefinitely. What is the expected number of bananas the monkey will eat?
Codex Input	A very innocent monkey throws a fair 6-sided die repeatedly. If the die value is 1 to 5, the monkey eats that many bananas and stops. If the die value is 6, the monkey will eat 5 bananas. Write a program that computes the average number of bananas that the monkey will eat before it stops.
Codex Output	<pre>import random  def main():     # ask user for number of trials     trials = int(input("How many trials? "))     # initialize the number of bananas eaten     bananas = 0     # loop for the number of trials     for i in range(trials):         # initialize the number of bananas eaten         bananas_eaten = 0         # loop until the monkey stops eating bananas         while True:             # generate a random number between 1 and 6             die = random.randint(1, 6)             # if the die value is 1 to 5, the monkey eats that             # many bananas and stops             if die &lt;= 5:                 bananas_eaten += die                 break             # if the die value is 6, the monkey will eat 5             # bananas             elif die == 6:                 bananas_eaten += 5             # add the number of bananas eaten to the total             bananas += bananas_eaten         # calculate the average number of bananas eaten         average = bananas / trials         # print the average number of bananas eaten         print("The average number of bananas eaten is",               average)  main()</pre>
Solution	4

Question 14 from STAT110

Topic	Conditional Expectation
Original Question	A fair coin is tossed repeatedly until 5 consecutive heads occurs. What is the expected number of coin tosses?
Codex Input	Write a program to simulate the average number of coin flips needed until you get 5 heads in a row for the first time.
Codex Output	<pre> import random  def flip_coin():     return random.choice(['H', 'T'])  def flip_until_five_heads():     heads = 0     count = 0     while heads &lt; 5:         count += 1         if flip_coin() == 'H':             heads += 1         else:             heads = 0     return count  def average_flips_until_five_heads():     total = 0     for i in range(1000):         total += flip_until_five_heads()     return total / 1000  print(average_flips_until_five_heads()) </pre>
Solution	62

Question 15 from STAT110

Topic	Geometric Probability, Expected Value
Original Question	A stick of length 1 breaks at a random point distributed uniformly across its length. What is the expected length of the smaller part of the broken stick?
Codex Input	A stick of length 1 is split into two portions at a random point distributed uniformly across its length. Write a simulation program that estimates the average length of the shorter portion of the stick.
Codex Output	<pre> import random  def stick_splitting(n):     """     :param n: number of simulations to run     :return: average length of the shorter portion of the              stick after splitting it in half n times.     """      total = 0      for i in range(n):          # generate a random number between 0 and 1,         # representing the point at which to split the         # stick.         split_point = random.random()          # generate a random number between 0 and 1,         # representing the length of the first portion of         # the stick.         first_portion = random.random()          # generate a random number between 0 and 1,         # representing the length of the second portion of         # the stick.         second_portion = 1 - first_portion          # add up all lengths from each simulation run to get         # an average length for each simulation run.         total += min(first_portion, second_portion)      return total / n  print(stick_splitting(100000)) </pre>
Solution	1/4

Question 16 from STAT110

Topic	Expected Value
Original Question	You are given an urn with 100 balls (50 black and 50 white). You pick balls from urn one by one without replacements until all the balls are out. A black followed by a white or a white followed by a black is 'a color change'. Calculate the expected number of colour changes if the balls are being picked randomly from the urn.
Codex Input	You are given an urn with 100 balls (50 black and 50 white). A sequence of balls is drawn from the urn without replacement until the urn is empty. Drawing a black ball immediately followed by a white ball is considered a color change. Drawing a white ball immediately followed by a black ball is also considered a color change. Write a simulation program that estimates the average number of color changes in a 100-ball sequence.
Codex Output	<pre> import random  def color_change(urn):     """Returns the number of color changes in a 100-ball     sequence."""     count = 0     while len(urn) &gt; 0:         ball = urn.pop()         if len(urn) == 0: # last ball, no color change             break          next_ball = urn[-1] # next ball in sequence          if (ball == 'B' and next_ball == 'W') or (ball == 'W'             and next_ball == 'B'): # color change!             count += 1      return count  def main():      urn = ['B'] * 50 + ['W'] * 50 # create an urn with 50         black and 50 white balls      trials = 10000 # number of trials to run for simulation      total_color_changes = 0 # initialize total number of         color changes to zero      for i in range(trials): # run the simulation for the         specified number of trials         random.shuffle(urn) # shuffle the balls in the urn             before each trial          total_color_changes += color_change(list(urn)) # add             the number of color changes in this trial to the             total      print(total_color_changes / trials) # print the average         number of color changes over all trials  if __name__ == '__main__':     main() </pre>
Solution	50

Question 17 from STAT110

Topic	Conditional Probability
Original Question	X and Y are i.i.d. $N(0, 1)$ random variables. You are given that $X > 0$ and $Y > 0$ . What is the probability that $Y > X$ ?
Codex Input	X and Y are i.i.d. $N(0, 1)$ random variables. Write a program to calculate the probability that $Y > X$ if both $X > 0$ and $Y > 0$ .
Codex Output	<pre> # X and Y are i.i.d. N(0, 1) random variables. # Write a program to calculate the probability that Y &gt; X   if both X &gt; 0 and Y &gt; 0.  import numpy as np import matplotlib.pyplot as plt from scipy import stats  def prob_Y_greater_X():      # Generate 1000 samples of X and Y, where X ~ N(0, 1)     and Y ~ N(0, 1).      x = np.random.normal(loc=0, scale=1, size=1000)     y = np.random.normal(loc=0, scale=1, size=1000)      # Calculate the probability that Y &gt; X if both X &gt; 0 and     Y &gt; 0      prob = sum([1 for i in range(len(x)) if x[i]&gt;0 and                 y[i]&gt;0 and y[i]&gt;x[i]])/sum([1 for i in range(len(x))                 if x[i]&gt;0 and y[i]&gt;0])      return prob  if __name__ == '__main__':      print(prob_Y_greater_X()) </pre>
Solution	1/2

Question 18 from STAT110

Topic	Conditional Expectation
Original Question	Suppose that you roll a dice. For each roll, you are paid the face value. If a roll gives 4, 5, or 6, you can roll the dice again. If you get 1, 2, or 3, the game stops. What is the expected payoff of this game?
Codex Input	Suppose we play a game where you roll a dice. If a roll gives 4, 5, or 6, you are paid the face value and you roll the dice again. If a roll gives 1, 2, or 3, you are paid the face value and you stop rolling. Write a simulation program that estimates the average payoff of this game.
Codex Output	<pre> import random  def roll_dice():     return random.randint(1,6)  def play_game():     total = 0.0     while True:         roll = roll_dice()         if roll == 1 or roll == 2 or roll == 3: # stop             rolling and get paid the face value of the dice             roll.             total += roll # add the face value of the dice to                 the total.             break # stop playing the game.         else: # keep rolling and add the face value             to the total.             total += roll      return total # return the total amount won from         playing this game.  # main program starts here, DO NOT change it. It runs the simulation and prints results. number_of_games = 10000 # number of times we play this game. Change it to 1000 to see more accurate results. total = 0 # initial condition - we don't have any money at hand right now. for i in range(number_of_games):     total += play_game() # add the amount won from each game to get the total amount won. print("The average payoff from {} games is {}".format(number_of_games, total/number_of_games)) </pre>
Solution	7

Question 19 from STAT110

Topic	Bayes' Theorem and Conditional Probability
Original Question	A spam filter is designed by looking at commonly occurring phrases in spam. Suppose that 80% of email is spam. In 10% of the spam emails, the phrase "free money" is used, whereas this phrase is only used in 1% of non-spam emails. A new email has just arrived, which does mention "free money". What is the probability that it is spam?
Codex Input	Suppose that 80% of email is spam. In 10% of the spam emails, the phrase "free money" is used, whereas this phrase is only used in 1% of non-spam emails. A new email has just arrived, which mentions "free money". Given that the email mentions "free money" and using Bayes' theorem, write a program to compute and print the probability that the email is spam.
Codex Output	<pre>def bayes_theorem(p_spam, p_free_money_given_spam,                   p_free_money_given_not_spam):     return (p_spam * p_free_money_given_spam) / (p_spam *   p_free_money_given_spam + (1 - p_spam) *   p_free_money_given_not_spam)  p_spam = 0.8 p_free_money_given_spam = 0.1 p_free_money_given_not_spam = 0.01  print(bayes_theorem(p_spam, p_free_money_given_spam,                     p_free_money_given_not_spam))</pre>
Solution	0.9756

Question 20 from STAT110



## H Columbia University COMS3251: Computational Linear Algebra

Topic	Matrix Algebra
Original question	Compute the following expression: $([-1, 0, 2; 0, 1, 4]^T * [-2; 1]).([3, 1; 0, 0; -2, -1] * ([5, -3] + [-4, 2]))'$ , where ' means transpose, * means matrix product, . means inner product.
Codex input	Compute the following expression: $(\text{transpose}([-1, 0, 2; 0, 1, 4]) * [-2; 1]).([3, 1; 0, 0; -2, -1] * \text{transpose}([5, -3] + [-4, 2]))$
Codex output	<pre>import numpy as np  a = np.array([[ -1, 0, 2], [0, 1, 4]]) b = np.array([[ -2], [1]]) c = np.array([[3, 1], [0, 0], [-2, -1]]) d = np.array([[5], [-3]]) e = np.array([[ -4], [2]])  print(np.dot(np.transpose(a), b)) print(np.dot(c, (d+e)))</pre>
Solution	4

COMS3251, Question 1: Original question, Codex input and output, and solution.

Topic	Vectors, Lengths and Dot products
Original question	Compute the squared $L_2$ norm of the vector $[1; -4; 2; 8; -1]$ .
Codex input	Original question
Codex output	<pre>import numpy as np  def compute_squared_L2_norm(vector):     return np.sum(np.square(vector))  vector = np.array([1, -4, 2, 8, -1]) print(compute_squared_L2_norm(vector))</pre>
Solution	86

COMS3251, Question 2: Original question, Codex input and output, and solution.

Topic	Solving Linear System of Equations
Original question	Find a solution to the following system of four equations: $4x - 2y + 8z + w = 3$ , $-8x + 10y + 3w = -2$ , $3x - 1y + 10z + 5w = -1$ , $2x + 2y + 9z - 2w = 8$ .
Codex input	Write a program to find a solution to the system of four equations: $4x - 2y + 8z + w = 3$ , $-8x + 10y + 3w + 2 = 0$ , $3x - 1y + 10z + 5w + 1 = 0$ , $2x + 2y + 9z - 2w = 8$
Codex output	<pre> from sympy import *  x, y, z, w = symbols('x y z w')  eq1 = 4*x - 2*y + 8*z + w - 3 eq2 = -8*x + 10*y + 3*w + 2 eq3 = 3*x - 1*y + 10*z + 5*w + 1 eq4 = 2*x + 2*y + 9*z - 2*w - 8  solve([eq1, eq2, eq3, eq4], [x, y, z, w]) </pre>
Solution	$x = 11/4, y = 97/44, z = -4/11, w = -15/22$

COMS3251, Question 3: Original question, Codex input and output, and solution.

Topic	Solving Linear System of Equations
Original question	A mining company has two mines. One day's operation at mine A produces ore that contains 10 metric tons of copper and 120 kilograms of silver, while one day's operation at mine B produces ore containing 6 metric tons of copper and 140 kilograms of silver. Let $a=[10;120]$ and $b=[6;140]$ . Then $a$ and $b$ represent the daily output of mines A and B, respectively. Suppose that the mining company operates mine A for $x$ days and mine B for $y$ days. Calculate how many number of days each mine should operate in order to produce 72 tons of copper and 1340 kilograms of silver.
Codex input	Let $a=[10;120]$ and $b=[6;140]$ . Write a program to compute positive $x$ and $y$ such that $x*a + y*b = [72;1340]$
Codex output	<pre> a = np.array([10, 120]) b = np.array([6, 140]) c = np.array([72, 1340])  # Solve for x and y in the equation x*a + y*b = c using # the numpy linear algebra function linalg.solve(a, b) x, y = np.linalg.solve(np.vstack((a, b)).T, c) print("x: ", x) print("y: ", y) </pre>
Solution	$x=3, y=7$

COMS3251, Question 4: Original question, Codex input and output, and solution.

Topic	Vectors, Lengths and Dot products
Original question	At noon, the minute and the hour hands of an analog clock coincide. What is the angle (in degrees) subtended by the minute and hour hands of an analog clock at 1:15?
Codex input	At noon, the minute and the hour hands of an analog clock coincide. Write a program to compute the angle (in degrees) subtended by the hands of an analog clock at 1:15
Codex output	<pre> def clock_angle(hour, minute):     hour_angle = (hour * 30) + (minute * 0.5)     minute_angle = (minute * 6)     angle = abs(hour_angle - minute_angle)     return angle  print(clock_angle(1, 15)) </pre>
Solution	52.5

COMS3251, Question 5: Original question, Codex input and output, and solution.

Topic	Matrix Algebra
Original question	For what matrix $R$ , the following matrix equation is satisfied? $R \begin{bmatrix} 1,0,0,0,0;1,1,0,0,0;1,2,1,0,0;1,3,3,1,0;1,4,6,4,1 \end{bmatrix} = \begin{bmatrix} 1,0,0,0,0;0,1,0,0,0;0,1,1,0,0;0,1,2,1,0;0,1,3,3,1 \end{bmatrix}$
Codex input	<code>A = np.matrix([[1,0,0,0,0],[1,1,0,0,0],[1,2,1,0,0],[1,3,3,1,0],[1,4,6,4,1]]) B = np.matrix([[1,0,0,0,0],[0,1,0,0,0],[0,1,1,0,0],[0,1,2,1,0],[0,1,3,3,1]])</code> "***** Solve $X \cdot A = B$ for $X$ *****"
Codex output	<pre># Solve X*A = B for X X = np.linalg.solve(A, B) print(X)</pre>
Solution	$\begin{bmatrix} 1,0,0,0,0;-1,1,0,0,0;0,-1,1,0,0;0,0,-1,1,0;0,0,0,-1,1 \end{bmatrix}$

COMS3251, Question 6: Original question, Codex input and output, and solution.

Topic	Rank, Basis and Dimension
Original question	Given a $d$ -dimensional non-zero vector $v$ , compute the rank of the matrix $v \cdot v^T$
Codex input	Given a $d$ -dimensional non-zero vector $v$ , write a program to compute the rank of the matrix $v \cdot \text{transpose}(v)$
Codex output	<pre>import numpy as np  def rank(v):     return np.linalg.matrix_rank(np.dot(v, v.T))</pre>
Solution	1

COMS3251, Question 7: Original question, Codex input and output, and solution.

Topic	Four Fundamental Subspaces of a Matrix
Original question	Compute the dimension of the null space of the following matrix. $\begin{bmatrix} 1,2,0,-1;-2,-3,4,5;2,4,0,-2 \end{bmatrix}$
Codex input	Write a program that finds the dimension of the null space of the matrix $\begin{bmatrix} 1,2,0,-1;-2,-3,4,5;2,4,0,-2 \end{bmatrix}$ . Use <code>scipy.linalg.null_space</code> . Get the <code>shape[1]</code> of the null space.
Codex output	<pre>import numpy as np from scipy import linalg  A = np.array([[1,2,0,-1],[-2,-3,4,5],[2,4,0,-2]])  print(linalg.null_space(A))  print(linalg.null_space(A).shape[1])</pre>
Solution	2

COMS3251, Question 8: Original question, Codex input and output, and solution.

Topic	Four Fundamental Subspaces of a Matrix
Original question	Compute the dimension of the left null space of the following matrix. $\begin{bmatrix} 1,2;2,4;3,6;4,8 \end{bmatrix}$
Codex input	Compute the dimension of the left null space of the matrix $\begin{bmatrix} 1,2;2,4;3,6;4,8 \end{bmatrix}$
Codex output	<pre>import numpy as np A = np.array([[1,2],[2,4],[3,6],[4,8]]) print(np.linalg.matrix_rank(A))</pre>
Solution	3

COMS3251, Question 9: Original question, Codex input and output, and solution.

Topic	Orthogonality and Projections
Original question	Compute the orthogonal projection of the vector $[-6;4]$ onto the line spanned by the vector $[3;2]$
Codex input	Compute the orthogonal projection of the vector $[-6;4]$ onto the line spanned by the vector $[3;2]$
Codex output	<pre>import numpy as np  # Define the two vectors v1 = np.array([3, 2]) v2 = np.array([-6, 4])  # Compute the orthogonal projection of v2 onto v1 and print it out proj = (np.dot(v1, v2) / np.dot(v1, v1)) * v1 print(proj)</pre>
Solution	$[-30/13;-20/13]$

COMS3251, Question 10: Original question, Codex input and output, and solution.

Topic	Least Squares Approximation
Original question	Find the least squares solution to the following matrix equation. $[0,1;1,1;2,1]*x=[6;0;0]$
Codex input	Write a program that finds the least squares solution to the matrix equation $[0,1;1,1;2,1]*x = [6;0;0]$
Codex output	<pre>import numpy as np A = np.array([[0, 1], [1, 1], [2, 1]]) b = np.array([6, 0, 0]) x = np.linalg.solve(A, b) print(x)</pre>
Solution	$[-3;5]$

COMS3251, Question 11: Original question, Codex input and output, and solution.

Topic	Eigenvectors and Eigenvalues
Original question	Compute the eigenvalue associated with with the eigenvector $[-684/721;228/721]$ of the matrix $[-6,3;4,5]$
Codex input	Write a program that finds the eigenvalue associated with with the eigenvector $[-684/721;228/721]$ of the matrix $[-6,3;4,5]$
Codex output	<pre>import numpy as np A = np.array([[ -6, 3], [4, 5]]) eigenvalues, eigenvectors = np.linalg.eig(A) print(eigenvalues) print(eigenvectors) print(eigenvalues[0]*eigenvectors[:,0]) print(eigenvalues[1]*eigenvectors[:,1])</pre>
Solution	$-7$

COMS3251, Question 12: Original question, Codex input and output, and solution.

Topic	Matrix Algebra
Original question	Compute the inverse of the following matrix: $[-1,-2;-2,0]$
Codex input	Compute the inverse of the following matrix: $[-1,-2;-2,0]$
Codex output	<pre>import numpy as np A = np.array([[ -1, -2], [-2, 0]]) print(np.linalg.inv(A))</pre>
Solution	$[0,-1/2;-1/2,1/4]$

COMS3251, Question 13: Original question, Codex input and output, and solution.

Topic	Determinants
Original question	Compute the determinant of the following matrix: [3,-4,5;0,-1,-5;5,-4,3]
Codex input	Compute the determinant of the following matrix: [3,-4,5;0,-1,-5;5,-4,3]
Codex output	<pre>import numpy as np matrix = np.array([[3,-4,5],[0,-1,-5],[5,-4,3]]) print(np.linalg.det(matrix))</pre>
Solution	56

COMS3251, Question 14: Original question, Codex input and output, and solution.

Topic	Special Matrices
Original question	Find the real numbers a, b and c so that the following matrix is symmetric. [0,a+b,c+2;a,2,c;4,a+b,4]
Codex input	Use sympy to find a, b, c so that [0,a+b,c+2;a,2,c;4,a+b,4] = transpose([0,a+b,c+2;a,2,c;4,a+b,4])
Codex output	<pre>import sympy  a, b, c = sympy.symbols('a b c') A = sympy.Matrix([[0, a + b, c + 2], [a, 2, c], [4, a + b, 4]]) B = A.transpose() print(sympy.solve([A[i] - B[i] for i in range(3)], (a, b, c)))</pre>
Solution	a=2,b=0,c=2

COMS3251, Question 15: Original question, Codex input and output, and solution.

Topic	Matrix Decomposition
Original question	Find an LU decomposition of the following matrix: [-1,-1,2;2,0,3;-3,2,-1]
Codex input	Find an LU decomposition of the following matrix: [-1,-1,2;2,0,3;-3,2,-1]
Codex output	<pre>import numpy as np A = np.array([[ -1, -1, 2], [ 2, 0, 3], [-3, 2, -1]]) L = np.eye(3) U = A for i in range(3):     for j in range(i+1, 3):         L[j, i] = U[j, i]/U[i, i]         U[j, :] = U[j, :] - L[j, i]*U[i, :] print(L) print(U)</pre>
Solution	[1,0,0;-2,1,0;3,-5/2,1]*[-1,-1,2;0,-2,7;0,0,21/2]

COMS3251, Question 16: Original question, Codex input and output, and solution.

Topic	Matrix Decomposition
Original question	Find an QR decomposition of the following matrix: [1,0,2;0,2,0;0,-1,1]
Codex input	Find an QR decomposition of the following matrix: [1,0,2;0,2,0;0,-1,1]
Codex output	<pre>import numpy as np A = np.array([[1, 0, 2], [0, 2, 0], [0, -1, 1]]) Q, R = np.linalg.qr(A) print(Q) print(R)</pre>
Solution	[1,0,0;0,2/sqrt(5),1/sqrt(5);0,-1/sqrt(5),2/sqrt(5)]*[1,0,2;0,sqrt(5),-1/sqrt(5);0,0,2/sqrt(5)]

COMS3251, Question 17: Original question, Codex input and output, and solution.

Topic	Diagonalization and Eigenvectors/Eigenvalues
Original question	Diagonalize the following matrix: [2,-3,0;0,-1,0;1,3,1]
Codex input	Write a program that diagonalizes the matrix [2,-3,0;0,-1,0;1,3,1]
Codex output	<pre>import numpy as np A = np.array([[2,-3,0],[0,-1,0],[1,3,1]]) print(A) eig_val, eig_vec = np.linalg.eig(A) print(eig_val) print(eig_vec) D = np.diag(eig_val) print(D) V = eig_vec print(V) V_inv = np.linalg.inv(V) print(V_inv) print(np.dot(V, np.dot(D, V_inv)))</pre>
Solution	$[-1,0,1;-1,0,0;2,1,1]*[-1,0,0;0,1,0;0,0,2]*[0,-1,0;-1,3,1;1,-1,0]$

COMS3251, Question 18: Original question, Codex input and output, and solution.

Topic	Matrix Algebra
Original question	Compute the trace of the following matrix: [3,8,-2;1,0,2;-2,-1,5]
Codex input	Compute the trace of the following matrix: [3,8,-2;1,0,2;-2,-1,5]
Codex output	<pre>import numpy as np matrix = np.array([[3,8,-2],[1,0,2],[-2,-1,5]]) print(matrix.diagonal().sum())</pre>
Solution	8

COMS3251, Question 19: Original question, Codex input and output, and solution.

Topic	Four Fundamental Subspaces of a Matrix
Original question	Which of the vectors $v=[3;-3;1]$ , $u=[1;-2;1]$ is an element of the nullspace of the following matrix: [1,2,-3;-1,-1,0;-2,-3,3]
Codex input	Write a program that checks if a vector is an element of the nullspace of a matrix Use the program to check if the vector [3;-3;1] is an element of the nullspace of the matrix [1,2,-3;-1,-1,0;-2,-3,3] Use the program to check if the vector [1;-2;1] is an element of the nullspace of the matrix [1,2,-3;-1,-1,0;-2,-3,3]
Codex output	<pre>import numpy as np  def is_nullspace(matrix, vector):     return np.allclose(np.dot(matrix, vector), 0)  if __name__ == '__main__':     matrix = np.array([[1, 2, -3], [-1, -1, 0], [-2, -3, 3]])     vector = np.array([[3], [-3], [1]])      print(is_nullspace(matrix, vector))      print(is_null([[1, 2, -3], [-1, -1, 0], [-2, -3, 3]],                   [1, -2, 1]))</pre>
Solution	$u=[1;-2,1]$

COMS3251, Question 20: Original question, Codex input and output, and solution.