

Quiz 1

Instructions:

- Read all the instructions on this page.
- You are allowed to access the internet as well as any files you may have on your computer. Feel free to copy solutions from the internet, but you **must cite anything you use**. This is at your own risk – if you copy a solution and it does not work, we will not give you points just because some website said it was correct.
- You will need to submit a single PDF file on gradescope, as well as some python files (in the usual way – as a zipped folder named Firstname_Lastname_Exam1.zip on the classroom). You should submit on gradescope first. That’s the time we will use to check if you submitted late.
- You must ALSO paste the code you write into the file you submit on gradescope. So you’ll submit the code twice – once pasted within your PDF, and again as a python file via classroom.
- The first page must contain:
 - Your name
 - Your Ashoka email
 - Today’s date (October 27, 2021)
 - The words “FC-0309 Quiz 1”
- Each solution (not sub-part) must start on a new page. (Paste the code for coding questions.)
- Just to make sure you’re actually reading the instructions: you’ll get a couple of extra points if you write your name twice (consecutively) just before the first question.
- You have 90 minutes. **Do not spend too much time on any single problem.** Read them all first, and attack them in the order that allows you to make the most progress. You are NOT required to answer all the questions. **I fully expect MOST of you to skip MULTIPLE questions!**
- If you are unable to upload everything on time, email your solutions to your TA.

Question	Points
Abstraction Art	25
Thinking About Thinking	25
Stranger Loops	25
Fibonacci Goes Forth	25
Sorting Things Out	25
Epicycloid Fun	25
Total:	150

Problem 1. [25 points] **Abstraction Art** (2 parts)

Statement: Some fields of knowledge lend themselves more naturally to abstraction (and encapsulation) than others. (2-3 paragraphs or so, not more than 1 page of text per subpart.)

- (a) [10 points] Does easier abstraction always lead to faster progress in a field? Why or why not?
- (b) [15 points] Which fields have the worst “abstractability” (name at least two) and why?

Problem 2. [25 points] **Thinking About Thinking** (2 parts)

Note that we expect you to refer to the things you learned while thinking about “Digital Infinity and The Computational Theory of Mind”, as well as the movie about Shannon.

- (a) [15 points] When humans find something difficult to do, we build a machine to help us.
If building a creative, “thinking” machine is too hard, could humans build a machine which builds better machines (which builds better machines, and so on), eventually leading to a true artificial intelligence? (3-4 paragraphs or so, not more than 1 page of text.)
- (b) [10 points] At the end of the movie “The Bit Player”, Shannon shows off his greatest invention, a useless machine: when the switch on the outside is turned on, a small hand comes out of the machine (essentially a box) and turns it off, then goes back inside. See: https://en.wikipedia.org/wiki/Useless_machine. (Many other results are easily Googled.)
What’s the big deal? Why are people like Claude Shannon, Marvin Minsky, and even science fiction authors like Arthur C. Clarke so fascinated by this machine? Or is it all just fluff and nonsense? I want to hear what you think, not just what other people have said. (2-3 paragraphs or so, not more than 1 page of text.)

Problem 3. [25 points] **Stranger Loops** (3 parts)

Debayan has decided to create a new kind of reading experience: He has written a book with thousands of pages, and given each page to a different person. Additionally, he has hidden away many more pages in various places across the world.

Each page has clues to THREE locations where some other page is stored. Note that each page is unique; no two pages point to the same three locations (partial overlaps are fine – a page can point to locations A,B,C, while another page could point to A,B,D). Pages can point to their own locations.

A reader can, if they choose, switch the page they are carrying for the page at the location where they currently are.

- (a) [15 points] Is this game a strange loop? Why or why not?
- (b) [5 points] What if each page had clues to TWO other locations?
- (c) [5 points] Could this be an infinite loop? How?

Problem 4. [25 points] **Fibonacci Goes Forth** (4 parts)

Recall the Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13,... each term is the sum of the last two terms. Feel free to find/copy code for this from the internet. The first two parts of this question should be trivial! Your solution to each subpart should be titled “4a.py”, “4b.py”, etc.

- (a) [5 points] Start with a variable “n”. Print the first n terms of the Fibonacci sequence.

- (b) [5 points] Start with a variable “ n ”. Print all terms of the Fibonacci sequence upto (but not including) value n .
- (c) [8 points] We want the same list as part (a), but only print every fourth item. This would look like: 2, 13, 89, 610, 4181,... Note that we will end up printing fewer terms than part (a).
- (d) [7 points] We want the same list as in part (a), but only print odd numbers. This would look like: 1, 1, 3, 5, 13, 21,... (Again, we print fewer terms.)

Problem 5. [25 points] **Sorting Things Out** (3 parts)

You should use the python files given to you on the classroom for quicksort. The edits below are meant to be **minimal**. The smaller/fewer changes you can make, the more points we'll give you! Your solution to each subpart should be titled “5a.py”, “5b.py”, etc.

- (a) [7 points] Edit the function partitionBetter so that (if we use this function to partition) we end up with a final output which is sorted in descending order (biggest to smallest).
- (b) [7 points] Edit the function partitionTerrible so that (if we use this function to partition) we end up with a final output which is sorted in descending order (biggest to smallest). You are NOT allowed to use the same change that you made in part (a)!
- (c) [11 points] Edit the function quicksort and partition (Terrible or Better, whichever you find easier) so that the list is broken into three parts and then rejoined, instead of two. That is, you want to have TWO pivots, which split the list into three subparts, which you then sort, join back up, etc.

Problem 6. [25 points] **Epicycloid Fun** (2 parts)

You should use the python files given to you on the classroom for epicycloids. Your solution to each subpart should be titled “6a.py”, “6b.py”, etc.

- (a) [10 points] Draw two epicycloids within one another. That is, one epicycloid must have an empty space within it, and the second epicycloid must be drawn in that space. This will require the second epicycloid to be smaller in size and also positioned appropriately!
- (b) [15 points] Have a new variable n . Now, draw n epicycloids, each starting from the vertex of an n -sided polygon, with each side=100. I don't care which epicycloids you draw, just that they start at the correct positions!