## Python for Math and Stat Fall 2022 Final Exam

This exam is worth 75 points. Assume that all necessary packages have been imported. When done with the exam, please **scan and upload to Gradescope**, then hand in the paper version.

1. (12 pts) Let

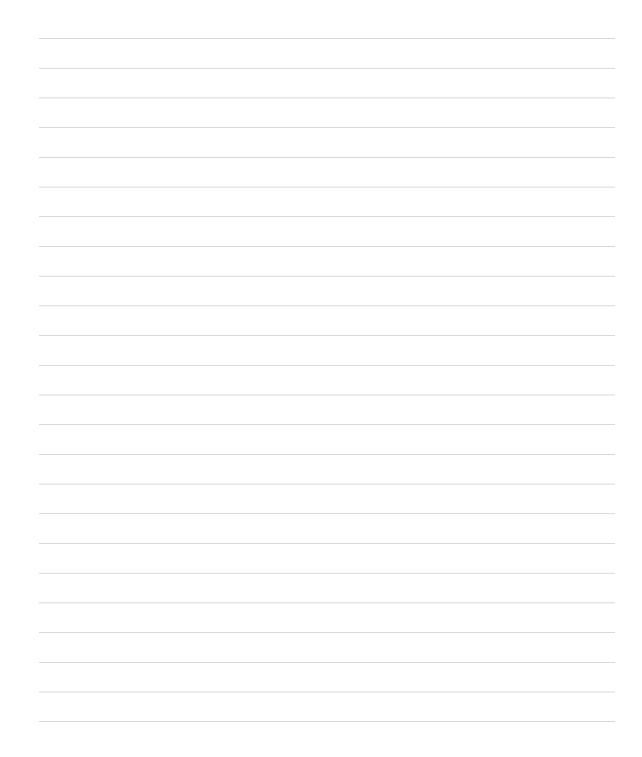
arr = np. array([[ 7, 13, 3, 2], [12, 6, 9, 5]])

For the following 4 problems, write down what each code block would display if executed in a Jupyter cell.

- (a) arr[1, ::-1]
- (b) arr[:, 3] \*\* 2
- (c) arr[arr // 10 > 0]
- (d) (lambda x: x+10)(arr[1, 2:])

2. (8 pts) Write a function di gi t\_i n\_num(di gi t, num) that returns True if the integer num contains di gi t and returns Fal se otherwise. Assume that num is a positive i nt and that di gi t is an i nt between 0 and 9 inclusive. Use arithmetic operations. DO NOT use string operations.

Examples: di gi t\_i n\_num(8, 56180) returns True. di gi t\_i n\_num(4, 5618073) returns Fal se.

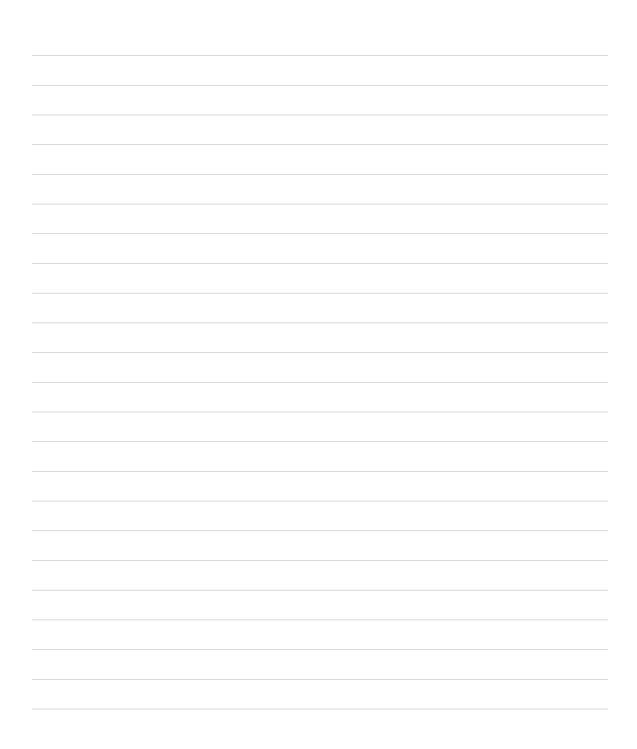


3. (8 pts) Write a function translate(words, lang\_dict) that takes a string of words, separated by spaces, and returns a new string, replacing each word found in lang\_dict with its equivalent. If a word is not found in the dictionary, it appears unchanged in the new string.

Example: To convert Spanish words into English, create a Spanish-English dictionary like the one below. Suppose the word ' pi ton' (which means python) is not included.

span\_dict = { 'casa': 'house', 'azul': 'blue', ... }

Then translate('casa azul piton', span\_dict) returns 'house blue piton'.



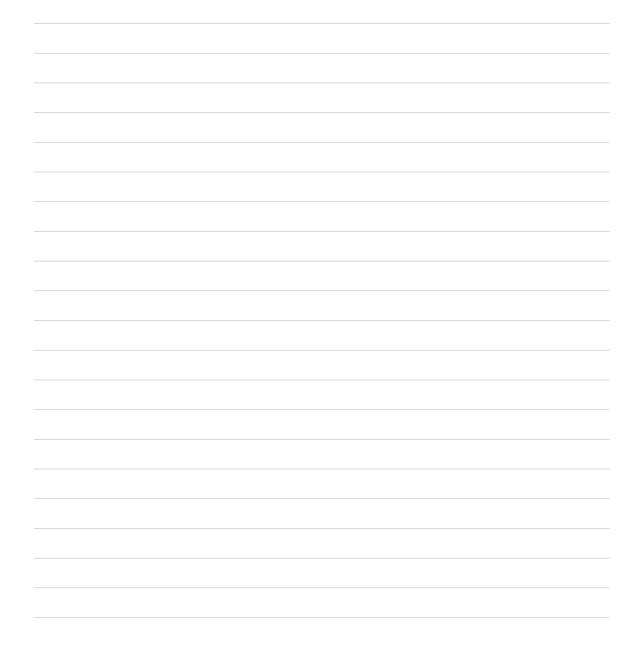
4. (8 pts)

```
def func(n):
    return ...
```

Suppose func is an increasing function and you wish to find a value of n such that func(n) is greater than a threshold value. Write a function exceed(thresh) that checks the integers n=1, 2, 4, 8, ..., one at a time, and stops when func(n) is greater than thresh, returning the successful value of n. Each iteration doubles the previous value of n. (Assume that the domain and range of func include all positive real numbers.)

## Example:

Suppose func(n) returns n + 2\*\*n. Then exceed(25) returns 8 because 4 +  $2^4$  < 25 and 8 +  $2^8$  > 25.

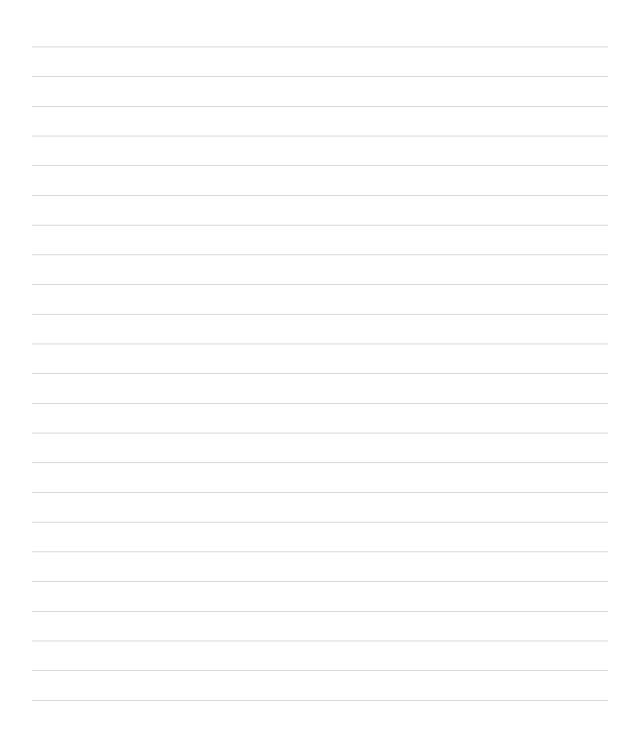


5. (8 pts) Consider the nested square root expression

 $a_1 + \overline{a_2 + \cdots + \overline{a_n}}.$ 

Write a **recursive** function **roots** (nums) that takes a non-empty list of positive numbers  $a_i$  and returns the value of the corresponding nested square root expression.

Example: roots([7, 2, 4]) returns 3. 0 because  $\overline{7 + \overline{2} + \overline{4}} = 3$ .



- 7. (19 pts)
  - (a) Create a class called **Point**. Each object in the class represents a point in the *xy*-plane. It has two attributes:
    - **x**: the *x*-coordinate of the point
    - **y**: the *y*-coordinate of the point
    - and the following methods:
      - dist(pt)

- (b) Add this Point method:
  - connect(pt\_l i st): given a list of Points, draws line segments connecting the given point to the other points, in order.

Example:

pt1 = Point(-2, 6)
pt2 = Point(3, 3)
Point(1, 2).connect([pt1, pt2])
draws line segments from (1, 2) to (-2, 6) to (3, 3).

