Short Answer Questions (40 points)

1. What is the Python type for representing decimal numbers?	float
2. Why aren't all decimal numbers are representable in Python?	Because Python uses a fixed (finite) number of bits to represent a decimal there are only a finite number of decimals that can be represented
3. What does the memory for a Python *variable* store? Be precise?	A variable stores a pointer/reference/id for a memory location storing an object
4. Suppose s is a variable. What does function invocation *type*(*s*) produce?	Type of s
5. What is does it mean that a function is built-in?	No import is necessary.
6. What is the value of the expression `12 % 7`?	5
7. What is the value of the expression `7 // 12`?	0
8. Suppose *s* points to string ' x y z '. What is the value of the expression *s.strip*()?	'x y z'
9. Suppose *s* points to string ' x y z '. What is the value of the expression *s.split*()?	['x', 'y', 'z']
10. Expression *range*(*a*, *b*) represents what sequence of values?	The integers a through b – 1
11. What is wrong with the expression `4 + 'th of July'`?	The + operator is not defined for a combination of int and string operands

12. Suppose variable *s* points to string `'answer'`. What is the value of expression *s*`[ 1 ]`?	'n'
13. Suppose variable *s* points to string `'answer'`. What is the value of expression *s*`[ 1 : 2 ]`?	'n'
14. Suppose variable s points to list `[ 'orange', 'blue' ]`. What is the value of expression *s.count*`( 'e' )`?	0 (there are no elements equal to 'e')
15. Suppose variable *s* points to string `'answer'`. What is the value of s after expression *s.upper*`()` is evaluated?	'answer' (upper() returns a new string in upper case format; it does not change s
16. Suppose variable *s* points to string `'answer'`. What is the value of expression *s.find*`( 'E' )`?	-1 (there is no location in s storing uppercase 'E'
17. Suppose variable *s* points to a string. How would you determine its length?	len(s)
18. Suppose variable *s* points to the list `[ '3', '14', '5' ]`. What is the value of expression *max*`( s )`?	'5' (it is last in *string* order)

19. What does the following code segment output?
`print( ':', end='-' )`
`print( '(', end=')' )`
	:-()

20. What is the value of variable *s* after the following code segment runs?
`s = 1`
`for x in range( 2, 4 ) :`
`  s = s * x`
	6

Part 2: Problem solving

21. (15 points) Implement program hoo.py whose only output is the string 'wahoo'. Your program header comment should provide name, email id, and program purpose. A program run is always

    wahoo

22. (15 points) Implement program chuck.py that separately prompts and gets two integer inputs: a number of woodchucks *w* and a number of days *d*. The program computes the number of cubic centimeters of wood, *w* woodchucks chuck over *d* days, where the amount of wood a single woodchuck can chuck in a day is 362 cubic centimeters.

    The only program output is to be the number of cubic centimeters of wood. Two sample program runs are

```
Enter number of woodchucks: 11
Enter number of days: 12
47784
```

```
Enter number of woodchucks: 2
Enter number of days: 3
2172
```

---

23. (15 points) Implement program grab.py that separately prompts for a string *s* and a list of integers *numbers*. The program displays the string formed by using the *numbers* as indices into *s*. For example, suppose *s* equals 'abcdefghijklmnopqrstuvwxyz-' and *numbers* equals [4, 8, 26, 4, 8, 26, 14] the output is the string 'ei-ei-o', because *s*[4] equals 'e', *s*[8] equals 'I', *s*[14] , equals 'o', and *s*[26] equals '-'. Two sample program runs are

```
Enter string: computer
Enter list of indices: 5 1 7
tor
```

```
Enter string: gosh look what the cat is up to now
Enter list of indices: 24 5 17 17 27 23 32 0
sleeping
```

    Suggestion: build up your answer by accumulating its characters one by one by looking into s.

24. (15 points) Develop program awesome.py. The program separately gets the name of a CSV data set of integer values. The data set is stored in web folder:

`http://www.cs1112.org/datasets/csv/test1/`

The data set does not have a header line. The program prints the sum of the max value for each line in the data set. For example, if the data set is `dataset1.csv`

```
12,5,22,6
6,9,8,17,16
1,13,16,2
8,23,7,6,6
5,8,5,22
```

The max values are respectively 22, 17, 16, 23, and 22. So the output is 100. Two sample runs are

```
Enter name of data file: dataset1.csv
100
```

```
Enter name of data file: dataset2.csv
48
```

Possible algorithm

- Import the library for accessing web files.
- Determine name of the CSV data web file.
- Determine the web location of the data file of interest.
- Get the contents of data web file.
- Strip the acquired contents of leading and trailing whitespace.
- Convert the acquired contents into a list of lines.
- Split each line in the list of lines into a list of integers.
- Sum together the maximum value from each line.

```
print( 'wahoo' )
```

```
CUBIC_CENTIMETERS_PER_DAY_PER_WOODCHUCK = 362

reply = input( 'Enter number of woodchucks: ' )
w = int( reply )

reply = input( 'Enter number of days: ' )
d = int( reply )

answer = CUBIC_CENTIMETERS_PER_DAY_PER_WOODCHUCK * w * d

print( answer )
```

```
reply = input( 'Enter string: ' )
s = reply


reply = input( 'Enter list of indices: ' )
reply = reply.split()

for n in reply :
    i = int( n )
    print( s[ i ], end='' )

println()
```

```
import url


WEB_FOLDER = 'http://www.cs1112.org/datasets/csv/test1/'
reply = input( 'Enter name of dataset: ' )
link = WEB_FOLDER + reply


data = url.get_contents( link )
data = data.strip()
data = data.split( '\n' )


maxes = []
for row in data :
    row = row.split( ',' )
```

```
    line = []
    for column in row :
        nbr = int( column )
        line.append( nbr )
    line_max = max( line )
    maxes.append( line_max )

total = sum( maxes )
print( total )
```

Name

Email id:

Pledge: On my honor, I pledge that I have neither given nor received help on this test.

Signature*:*

Notice

- Based on your past educational achievements, I expect you will do well on this test.

Test rules

- You may use a single piece of paper as scrap.
- This pledged exam is closed notes. The only device you may access during the test is your laptop.
- Do not access class examples or your own past assignments during the test; that is, the only code you may access or view are ones that you develop for this test.
- The only windows to be open on your computer are PyCharm and a single browser with tabs only open to the class website.
- PyCharm can be used for developing the programs to be submitted. It cannot be used for the short answer questions.

- Programs should demonstrate follow style rules; e.g., header comments, whitespace, identifier naming, etc.
- Whether a program is runnable is important.
- Only output what is requested.
- Whenever a written answer is a string, surround the string with single quotes. Also, make sure the uppercase and lowercase letters are easily distinguishable.
- Whenever a written answer is a list, surround the elements with a pair of brackets.
- Whenever a written answer is a decimal, include a decimal point.
- Whenever a written answer is an integer, do not include a decimal point.
- Any form of cheating on a test can result in expulsion from the class and the incident being referred to the Honor Committee.