BRAC University

Department of Computer Science and Engineering

CSE110: Programming Language I

Examination: *Theory Assignment #1* Semester: *Summer 2022*

Date: __ __ / __ __ / 2022 Deadline: 8 August, 2022

ID: _ _ _ _ _ _ _ _ _	Name: *(Please write in CAPITAL LETTERS)*	Section: **29**

1. Coming from a JAVA background, your friend is having difficulties following the snake_case variable naming convention in Python. For you, it's not a very big deal, in fact, it's extremely easy for you as you are very much familiar with both snake_case and camelCase. Therefore, you have planned to write a Python program for your friend that will take the input of a one-word string as camelCase and convert it into snake_case and print the result.

   However, you have decided to give a surprise to your friend that your program will also inform if a given variable name is valid or invalid in Python. You already know the variable naming rule in Python which there can be more than one character from the range [A - Z | a - z | 0 - 9 | _] but it cannot begin with a number [0 - 9], you just need to write a few lines of Python code to check and inform your friend if he ever makes a mistake.

Sample Input	Sample Output
already_snake_case	already_snake_case
first name	Invalid variable name!
firstName	first_name
thisIsATestVariable	this_is_a_test_variable
1stVariable	Invalid variable name!

2. Although you love Algebra, you often find it difficult to remember the four basic algebraic identities which are:

   $$(a + b)^2 = a^2 + 2ab + b^2$$
   $$(a - b)^2 = a^2 - 2ab + b^2$$
   $$(a + b)(a - b) = a^2 - b^2$$
   $$(x + a)(x + b) = x^2 + x(a + b) + ab$$

   There are a lot of application and calculators that could help you but since you are learning to write programs using Python and you love to solve real-life problems with programming, you have decided to

write a Python program that will take input of the left-hand side of these equations then calculate and print the result of the right-hand side.

Additionally, you decided to handle any number of white spaces for which you will first remove all the white spaces from the given input. Since the program is already doing a lot of interesting stuff, you also included a few lines of code to check if the equation is a valid equation of the above 4 identities.

Sample Input	Sample Output
(12 + 3)(12 - 3)	Executing equation: (a+b)(a-b) Result: 135
(7 + 9)^ 2	Executing equation: (a+b)^2 Result: 256
(11 - 5)^2	Executing equation: (a-b)^2 Result: 36
(12+3)(12+4)	Executing equation: (a+b)(a+c) Result: 240
(9 + 1)(10 + 2)	Executing equation: (a+b)(c+d) Invalid! Please enter a valid LHS equation of 4 fundamental algebraic equations.
(12 + 12)(12 - 12)	Executing equation: (a+b)(a-b) Result: 0

3. Your sibling asked you to create a game for two players. Since it's Eid, you could not refuse so, you have decided to create an "ASCII Fighter". However, you are still not an expert in programming so, you have requested a guideline from your teacher and he sent you the following message:

   "You need to write a Python program that read the name of the first and second player first. Next, it read 10 string inputs with alternating turns - starting with player one. For each input, calculate the summation of ASCII values of each character. With every two turns, the player with the highest ASCII sum will get the point. Finally, print the winner of the game; if it's a draw, then print "That's a win-win!".

Sample Input	Sample Output
Name of Player One: John Name of Player Two: Beck Word for Player One: CSE Word for Player Two: 110 Word for Player One: CSE Word for Player Two: 111 Word for Player One: CSE Word for Player Two: 220 Word for Player One: CSE Word for Player Two: 221 Word for Player One: CSE Word for Player Two: 331	Congratulations John, you won by 5 scores.

Name of Player One: John Name of Player Two: Beck Word for Player One: T Word for Player Two: h Word for Player One: a Word for Player Two: n Word for Player One: k Word for Player Two: y Word for Player One: o Word for Player Two: u Word for Player One: ! Word for Player Two: !	Congratulations Beck, you won by 4 scores.
Name of Player One: John Name of Player Two: Beck Word for Player One: a Word for Player Two: a Word for Player One: b Word for Player Two: b Word for Player One: 15 Word for Player Two: f Word for Player One: 27 Word for Player Two: 36 Word for Player One: cse Word for Player Two: 222777	It's a win-win

4. Trace the following code.

```
1  i = 0
2  k = 0
3  l = 0
4
5  while i < 10:
6      j = i + 1
7      while j < 20:
8          l = i + j - (5 // 2.0) % 2 ** 3 // 2
9          k = i - j + int(l)
10         i += 1
11         print(k)
12         if k < 2:
13             j += 1
14         else:
15             j += 5 % 7
16     i += 1
17 print(l)
```

5. Trace the following code.

```
1  n = 6
2
3  for i in range((2 * n - 1) ** 2):
4      i, j = i // (2 * n - 1), i % (2 * n - 1)
5      a = i < n and j >= n - i - 1 and j <= n + i - 1
6      b = i >= n and j >= i - n + 1 and j <= 2 * n - 2 - (i - n)
7      c = n % 2 and i % 2 == j % 2 or not(n % 2) and i % 2 != j % 2
8      if (a or b) and c:
9          if i % 2:
10             print(end="#")
11         else:
12             print(end="*")
13     else:
14         print(end=" ")
15     print(end=(j == 2 * n - 2 or "") and "\n")
```

6. A conditional for-loop that requires a boolean value or expression to iterate does not exist in Python. However, since a lot of programmers feel comfortable with for-loops, therefore, Python included a list-like object known as `**range**`. Upon checking how it works, you have found that you can write create a range object only in three ways: range(end), range(start, end), and range(start, end, step); where `end` is a necessity parameter/ value and default value for `start` is 0 and step is (+)1. Since you try to avoid built-in objects and functions, you have decided to write a Python program to create a list from the input of range functions. To ensure you understood everything, you have decided to not use the following functions: `range(...)`, `list(...)`, `eval(...)`, `exec(...)`. Please show your program.

Sample Input	Sample Output
range(10)	`List: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]`
range(1, 10)	`List: [1, 2, 3, 4, 5, 6, 7, 8, 9]`
range(1, 10 , 2)	`List: [1, 3, 5, 7, 9]`
range(10 , 1)	`List: []`
range("hello")	`Invalid!`
range(9, 2, -3)	`List: [9, 6, 3]`

range(-8, -10, -1)	`List: [-8, -9]`

7. Since you heard from your teacher that it is better to avoid as many built-in functions as possible, you have been wondering if it is possible to convert a string, that only has numbers and is a valid integer string, to int. After learning the ASCII table, you feel that it is actually possible and hence, you asked your teacher for confirmation, and he replied, “YES!”. Therefore, you have decided to write a Python program that will read an input - as a string - from the user and convert it into an integer value if it is a valid integer value. To make sure your value is of int type, you need to use the `type(...)` function.

   As you are replicating the built-in `int(...)` function, in your program you will never use the built-in `int(...)` function.

Sample Input	Sample Output
000123	123 <class 'int'>
-123.0	Invalid!
12-3	Invalid!
+4564	4564 <class 'int'>
123000-	Invalid!
132464-4654	Invalid!
-5465645665	-5465645665 <class 'int'>
-00046450656	-46450656 <class 'int'>

8. After being able to complete task 7, which converts an integer string to int without the built-in `int(...)` function, you are thinking if you can convert an integer value to a string without the `str(...)` function. Well, good news! You can convert an integer to a string, again using the ASCII functions `ord(...)` and `chr(...)`.

Sample Input	Sample Output
123	123 <class 'str'>
+110	110 <class 'str'>
-110	-110 <class 'str'>
0	0 <class 'str'>