COL100 Assignment 3
Holi Semester : 2021-2022

Deadline: 11:59 PM, 16 April, 2022

General Instructions

You should attempt this assignment without taking help from your peers or referring to online resources except for documentation (we will perform a **plagiarism check** amongst all submissions). Any violation of above will be considered a breach of the honor code, and the consequences would range from **zero marks** in the assignment to a **disciplinary committee action**.

Submission Instructions

1. If you are solving the i^{th} question, code in a file named `qi.py`. For eg. solution code for the 2^{nd} question goes inside a file named `q2.py`.
2. Submit your code in a `.zip` file named in the format `<EntryNo>.zip`. Make sure that when we run unzip `<EntryNo>.zip`, a folder `<EntryNo>` should be produced in the current working directory. For eg. if your entry number is `2021CS5XXXX`, then your zip file would be `2021CS5XXXX.zip` and upon unzipping, *it should produce a folder* `2021CS5XXXX` containing files `q1.py`, `q2.py`, `q3.py` and so on. For reference, we would be uploading a sample zip file on Piazza containing the exact directory structure which is mentioned.
3. Your submissions will be **auto-graded**. Make sure that your code follows the specifications (including directory structure, input/output, submission `.zip` file) of the assignment precisely. Not following this can lead to marks cut. As a precautionary measure - after submission, you can download your submission and see that directory structure is maintained and programs are working.

Some Clarifications

1. Every problem description is followed by some examples showing how exactly input and output is being expected. Please refer to them for more clarity.
2. Make sure if output is integer, then it is printed as integer, i.e. without ".0" and if output is float then print it as float upto 2 decimal places. Make sure to "round off" to 2 decimal places for float output. **Do not "truncate" float outputs**.
3. For input if it is integer, then it is taken as integer, i.e. without ".0" and if input is float then take it as float with no trailing 0 . Also, every input should be in new line (in case of multiple input).
4. If a problem involves printing a floating point number, your output is considered correct as long as it is within 2 decimal places and follows the rule of rounding off. For eg. if you output a floating point number 1.26, and the actual answer is 1.25678, then your output is correct.
5. If you still have any more doubts, feel free to shoot them at Piazza.

1 Maximum Number (In-Lab Component)(5 marks)

This question is the combination of if-else and the variables. Your task is to write a python program which takes 3 numbers as input and find the max among the 3 using if-else statements only. You can not use max function anywhere and data structures like arrays, lists. Finally print the maximum of the three.

Your program should prompt the user to enter the three numbers, namely a, b and c respectively.

Example 1:
INPUT:

```
1
2
3
```

OUTPUT:

```
3
```

EXPLANATION:
Here in this example we are provided with three numbers, $a = 1, b = 2$ and $c = 3$. So, maximum of the three numbers is 3

Example 2:
INPUT:

```
1
-1.58
-6
```

OUTPUT:

```
1
```

Example 3:
INPUT:

```
1
1
0
```

OUTPUT:

```
1
```

Note: a, b, c can be any real number (i.e. need not only be integers).

2 Modified Calculator (5 marks)

Write a python program to perform the following binary operations on two integers a and b with given parameter p.

This time you are given a parameter as well in the input and you have to perform appropriate operation according to the parameter. Parameter p can take values $1, 2, 3, 4$ and 5 only, where each number refer to the following binary operation:

1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Modulus

Your program should prompt the user to enter the two numbers, namely a and b (You can assume that $b \neq 0$) and a parameter p and perform the operation according to the parameter. For e.g. if $p = 1$ then you have to print the addition of a and b i.e. $a + b$, if $p = 2$ then you have to print the subtraction of a and b i.e. $a - b$ and so on.

Example 1:
INPUT:

```
2
4
1
```

OUTPUT:

```
6
```

EXPLANATION:
Here in this example we are provided with three numbers, $a = 2$ and $b = 4$ and $p = 1$. So, from the list above 1 stands for addition. So, there addition will be $a + b = 6$.

Example 2:
INPUT:

```
2
4
2
```

OUTPUT:

```
-2
```

Example 3:
INPUT:

```
2
4
3
```

OUTPUT:

```
8
```

Example 4:
INPUT:

```
2
4
4
```

OUTPUT:

```
0.5
```

Example 5:
INPUT:

```
2
4
5
```

OUTPUT:

```
2
```

Note: It is guaranteed that p will not take values other than $1, 2, 3, 4$ and 5. Also, a and b are not necessarily integers, and can take any real value.

3 Quadratic Equation Solver (5 marks)

Write a python program to calculate the roots of a real quadratic polynomial. You may use the quadratic formula (Shri Dharacharya Formula) to solve this.

Given to us a real quadratic polynomial $p(x) = ax^2 + bx + c$ where $a, b, c \in R, a \neq 0$. The python program takes in the values of a, b and c respectively in its input and prints the two roots in the

standard output in $x + iy$ (where $i = \sqrt{-1}$) form, where x and y are space separated. Each root is in one line (printing order of the roots does not matter). Your program should prompt the user to enter the three numbers, namely a, b and c respectively. You are only allowed to use $sqrt()$ function for result calculation apart from binary operator.

Example 1:
INPUT:

```
1
2
1
```

OUTPUT:

```
-1 0
-1 0
```

EXPLANATION:
Here in this example we are provided with three numbers, $a = 1, b = 2$ and $c = 1$. So, $p(x) = x^2 + 2x + 1$, hence the two roots are same and both are $-1 + 0i$. Also note how in both the roots, x and y i.e. the real and the imaginary part of each root is space separated; and there is a different line for each root.

Example 2:
INPUT:

```
1
0
1
```

OUTPUT:

```
0 1
0 -1
```

EXPLANATION:
$p(x) = x^2 + 1$. Therefore, roots are $\pm i$.

Example 3:
INPUT:

```
1
2
3
```

OUTPUT:

```
-1 1.41
-1 -1.41
```

Note: a, b, c can take any real value. Space between real and imaginary part should be of 1 unit and both should follow output rule (given at top) individually.

4 Check Palindrome (5 marks)

Write a program to check whether a string is a palindrome or not. The string will be provided as input and you have to use if-else statements to check whether the string is a palindrome or not. *A palindrome is a string which reads the same backward as forward.* The string *madam* when spelt backwards is *madam* therefore is a palindrome however the string *check* when spelt backwards is *kcehc* which is not the same and is thus not a palindrome.

Your program should prompt the user to input a string. The length of the string will be less than or equal to 4 and length will **not** be provided, you can use the function $len(str)$ to find the same. You **cannot** use loops, in-built string functions or the slice statement $str[::-1]$ to solve this problem. You are **only** allowed to use the *len() function, if-else statements and string indexing.* The output will be "NO" if the string is not a palindrome and "YES" if the string is a palindrome.

Note: The output must be in capital letters only. The input can contain numbers, letters and special characters. Capital and small letters are considered as different i.e. "a" and "A" are not the same.

Example 1:
INPUT:

```
loop
```

OUTPUT:

```
NO
```

EXPLANATION:
The string loop when spelt backwards reads as pool which is not the same as loop and is thus not a palindrome.

Example 2:
INPUT:

```
aba
```

OUTPUT:

```
YES
```

EXPLANATION:
The string aba when spelt backwards reads as aba which is the same and is a palindrome.

Example 3:
INPUT:

```
a,b
```

OUTPUT:

```
NO
```

EXPLANATION:
The string a,b when spelt backwards reads as b,a which is not the same as a,b and is thus not a palindrome.

Example 4:
INPUT:

```
atta
```

OUTPUT:

```
YES
```

EXPLANATION:
The string atta when spelt backwards reads as atta which is the same as forwards.

Example 5:
INPUT:

```
Aa
```

OUTPUT:

```
NO
```

EXPLANATION:
The string forwards is Aa and backwards is aA, as both are different it is not a palindrome.