COL100 Assignment 4
Holi Semester : 2021-2022

Deadline: 11:59 pm, 23 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. You are **not** required to submit the lab question mentioned in the assignement as "In Lab Component". The question evaluation for a lab will be done in the lab only and evaluations not done for the in-lab component for a student in his/her lab slot will be marked 0. It is your duty to get them evaluated.
3. 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 example, 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.
4. Your submissions will be **auto-graded**. Make sure that your code follows the specifications (including directory structure, input/output, importing libraries, submission `.zip` file) of the assignment precisely.

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. Irrespective of whether the output is float or integer, always print the output rounded off to two decimal places. **Do not "truncate" outputs**. Please refer to Piazza for more details.
3. If you still have any more doubts, feel free to shoot them at Piazza.

1 Divisible Factorial (5 marks) (In lab Component)

Write a program to calculate the divisible factorial of a number using loops. The difference here is that you only multiply numbers divisible by certain numbers. Thus the divisible factorial should be multiplication of numbers which are divisible by a specific number x. You can use any loop in your program, including for loops, while loops, and do-while. Since a normal factorial would be $n! = 1 * 2 * 3 ... (n-1) * n$ but, the expected divisible factorial would be $n! = 2 * 4 * 6 * 8 ... (n-1) * n$ if the dividing number given is 2 and would be $n! = 3 * 6 * 9 * 12 ... (n-1) * n$ if the dividing number is 3.

Input: Your program should prompt the user to enter the numbers, named a and x.

Output: Print the factorial of a.

Notes:

1. User may input two numbers after a, then in such case factorial is multiplication of those numbers which are divisible by both these input numbers. There will only be a max of two numbers.
2. If none of the numbers in the factorial are divisible by the given number the answer would be 1.

Example 1:
INPUT:

```
1
1
```

OUTPUT:

```
1.00
```

EXPLANATION:
The factorial of 1 is equal to 1

Example 2:
INPUT:

```
5
2
```

OUTPUT:

```
4.00
```

EXPLANATION:
The factorial of 5 is equal to 4 because only 4 is divisible by 2 and 4 both. Usually $5! = 1 * 2 * 3 * 4 * 5$ but out of these numbers only 4 is divisible by 2 and 4 both thus $5! = 4$

Example 3:
INPUT:

```
10
5
```

OUTPUT:

```
50.00
```

EXPLANATION:
The factorial of 10 is equal to 50 because 5, 10 are only divisible by 5. Usually $10! = 1 * 2 * 3 * 4 * 5 * 6 * 7 * 8 * 9 * 10$ but we want to multiply numbers divisible by 5 thus $10! = 5 * 10$

Example 3:
INPUT:

```
6
17
```

OUTPUT:

```
1.00
```

EXPLANATION:
Here the answer is 1 as none of the numbers in the factorial are divisible by the specified number. Usually $6! = 1 * 2 * 3 * 4 * 5 * 6$ but none of these are divisible by 17 and the answer would be $6! = 1$.

2 Pattern Printing - 1 (5 marks) (In Lab Component)

Write a python program which takes a positive number as input and print the pattern as shown in the test cases (line spacing doesn't matter). The input indicates the number of lines to be printed.
The pattern is a hollow square and each line must have the same number of characters i.e. the lines apart from the first and the end will have spaces in between. The number of lines and the number of characters in a line are the same.

Note: Here it looks more like a rectangle due to improper line spacing and line width. You need not worry about line spacing or width in the output.

Example 1:
INPUT:

```
5
```

OUTPUT:

```
*****
*   *
*   *
*   *
*****
```

Example 2:
INPUT:

```
6
```

OUTPUT:

```
******
*    *
*    *
*    *
*    *
******
```

3 Pattern Printing - 2 (10 Marks)

Write a python program which takes a positive number as input and print the pattern as shown in the test cases. Each line has an odd number of stars and the number of lines where the input increases is equal to n where n is the input given.

Example 1:
INPUT:

```
3
```

OUTPUT:

```
*
***
*****
***
*
```

Example 2:
INPUT:

```
4
```

OUTPUT:

```
*
***
*****
*******
*****
***
*
```

Example 3:
INPUT:

```
5
```

OUTPUT:

```
*
***
*****
*******
*********
*******
*****
***
*
```

4 Pattern Printing - 3 (10 Marks)

Write a python program which takes a positive number as input and print the pattern as shown in the test cases. The number of characters in a line and the number of lines are equal to each other and to the input given.

Note: The input number's range is 1-26 only. The ouput must be in capital letters and outputs as aaa or bbb would not be accepted.

Example 1:
INPUT:

```
3
```

OUTPUT:

```
AAA
 BB
  C
```

Example 2:
INPUT:

```
4
```

OUTPUT:

```
AAAA
 BBB
  CC
   D
```

Example 3:
INPUT:

```
5
```

OUTPUT:

```
AAAAA
 BBBB
  CCC
   DD
    E
```

Example 4:
INPUT:

```
1
```

OUTPUT:

```
A
```

5 Pattern Printing - 4 (15 Marks)

Write a python program which takes a positive number as input and print the pattern as shown in the test cases. The given input indicates the number of lines to be printed. Note that there is a space after each integer in a line.

Example 1:
INPUT:

```
3
```

OUTPUT:

```
1
2 3
4 5 6
```

Example 2:
INPUT:

```
4
```

OUTPUT:

```
1
2 3
4 5 6
7 8 9 10
```

Example 3:
INPUT:

```
5
```

OUTPUT:

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

6 Calculate the Terms of given Series (15 Marks)

Calculate the terms of a series considering the series be: $\frac{(-1)^n}{x^{2n}}$, and some terms of the series be: $\frac{-1}{x^2}, \frac{1}{x^4}, \frac{-1}{x^6}, \frac{1}{x^8} \cdots$

Input: Two numbers for x and n, where x is the value and n is the number of terms you have to print.

Output: Printing the terms of the given series. Don't print the sum of the terms.

Note: You may use the *pow*() function or the *math.pow*() function to find the powers.

Example 1:
INPUT:

```
1
3
```

OUTPUT:

```
-1.00
1.00
-1.00
```

EXPLANATION:
The first term of the given series $\frac{(-1)^n}{x^{2n}}$ is -1 on putting n=1 and the second is 1 on putting n=2 and Similarly the third term is -1 on putting n=3.

Example 2:
INPUT:

```
3
2
```

OUTPUT:

```
-0.11
0.01
```

EXPLANATION:
Here $x = 3$ and $n = 2$ putting them in the series gives the following output.

Example 3:
INPUT:

```
-2
2
```

OUTPUT:

```
-0.25
0.06
```

Please take a note of the sign of the number since any term also includes the sign.