Lab: Basic Syntax, Conditional Statements and Loops

Problems for in-class lab for the Python Fundamentals Course @SoftUni. Submit your solutions in the SoftUni judge system at https://judge.softuni.bg/Contests/1718

1. Biggest of Three Numbers

Write a program that receives **three whole numbers** and print the **biggest one**

Example

Input	Output
3 -1 5	5
0 -1 -2	0

Hints

We start by reading the three numbers from the console:

biggest-of-three-numbers.py

```
first_num = int(input())
second_num = int(input())
third_num = int(input())

```

Then we compare them and print the biggest one:

```
if first_num > second_num and first_num > third_num:
    print(first_num)
elif second_num > first_num and second_num > third_num:
    print(second_num)
else:
    print(third_num)

```

2. Number Definer

Write a program that reads a floating-point number and prints **"zero"** if the number is zero. Otherwise, print **"positive"** or **"negative"**. Add **"small"** if the absolute value of the number is less than 1, or **"large"** if it exceeds 1 000 000.

Example

Input	Output
25	positive

0.7	small positive
435247392.921	large positive
-0.005	small negative
-103.21	negative
-358583355123.001	large negative

Hints

First, we read the number from the console as a float, because we are going to receive floating point numbers:

```
1    number = float(input())
```

Then, we write the conditions if the number is zero:

```
2    if number == 0:
3        print("zero")
```

After that, we write the condition if the number is positive and add the additional conditions to that:

```
4    elif number > 0:
5        if number < 1:
6            print("small positive")
7        elif number > 1000000:
8            print("large positive")
9        else:
10           print("positive")
```

Then, if the number is negative. For the nested conditions here, we use the absolute value of the number, because that is what we need:

```
11   else:
12       if abs(number) < 1:
13           print("small negative")
14       elif abs(number) > 1000000:
15           print("large negative")
16       else:
17           print("negative")
```

3. Word Reverse

Write a program that receives a single word from the user, reverses it and prints it

Example

Input	Output
Python	nohtyP
banana	ananab

Hints

We read the word from the console

```
1    word = input()
```

We create new variable to store the reversed word

```
2    reversed_word = ""
```

We create a reversed loop and add each character to the new word

```
3    for i in range(len(word) - 1, -1, -1):
4        reversed_word += word[i]
5    print(reversed_word)
```

- We start form the length of the word - 1, because indices start from 0
- To loop until we reach 0, we type as end -1, because the end is not inclusive
- Finally the step is negative, so we type -1

4. Number Between 1 and 100

Write a program which reads numbers from the console until it receives a number between 1 and 100 inclusive. When the correct number is received, stop reading and print **"The number {number} is between 1 and 100"**

Example

Input	Output
-3 0.9 44	The number 44.0 is between 1 and 100

Hints

We start by reading a floating-point number

```
1    number = float(input())
```

We create a while loop with a condition for the number being less than 1 or greater than 100, since that is when we want to enter the body of the loop:

```
2    while number < 1 or number > 100:
3        number = float(input())
```

If the condition is not met, we exit the loop and print the result:

```
4    print(f'The number {number} is between 1 and 100')
```

5. Patterns

Write a program to create the following pattern:

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

You will receive a number that represents the highest number of stars.

Example

Input	Output
3	* ** *** ** *
5	* ** *** **** ***** **** *** ** *

Hints

First, we read the number from the console:

```
number = int(input())
```

We create the first loop, which will print the half of the pattern, until **`i == number`**:

```
for i in range(1, number + 1):
    for j in range(0, i):
        print('*', end='')
    print()
```

We create the inner loop to loop through the number from **`0`** to **`i`** (remember **`i`** is not inclusive). We use **`end=''`** to stay on the same line. We print the new line after we draw all the start for the particular line

Now, we create the second loop, which will draw one line less than the previous one (since the count of the lines is always odd):

```
for i in range(number - 1, 0, -1):
    for j in range(0, i):
        print('*', end='')
    print()
```

Here we loop backwards, because we decrease the number of stars with each line.