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.](https://judge.softuni.bg/Contests/1718)

1. Largest of Three Numbers

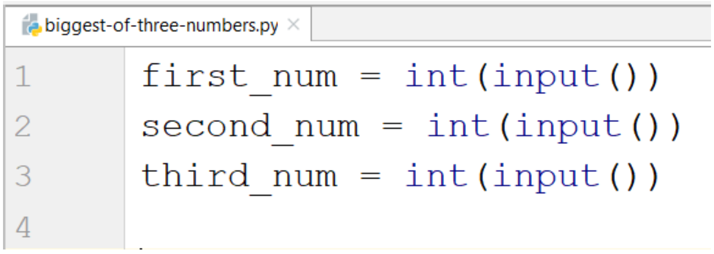
Write a program that receives **three whole numbers** and print the **largest one.**

Example

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

Hints

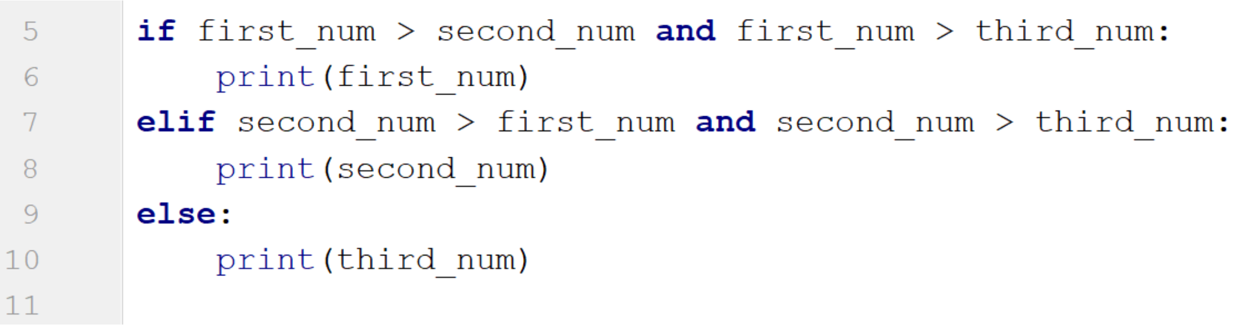
We start by reading the three numbers from the console:



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

```

Then we compare them and print the largest one:



```python
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, it should print **"positive"** or **"negative"**. The program should add **"small"** if the absolute value of the number is less than 1, or **"large"** if it exceeds 1 000 000.

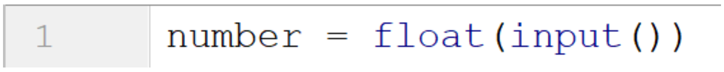
Example

Input	Output
25	positive

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

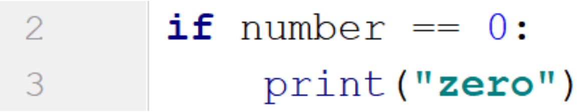
Hints

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



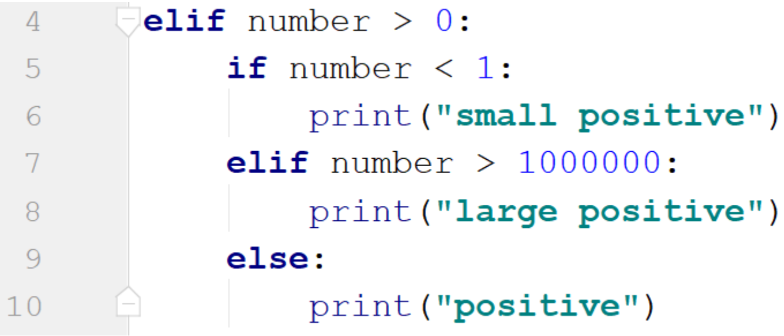
```python
number = float(input())
```

Then, we write a condition to check if the number is zero:



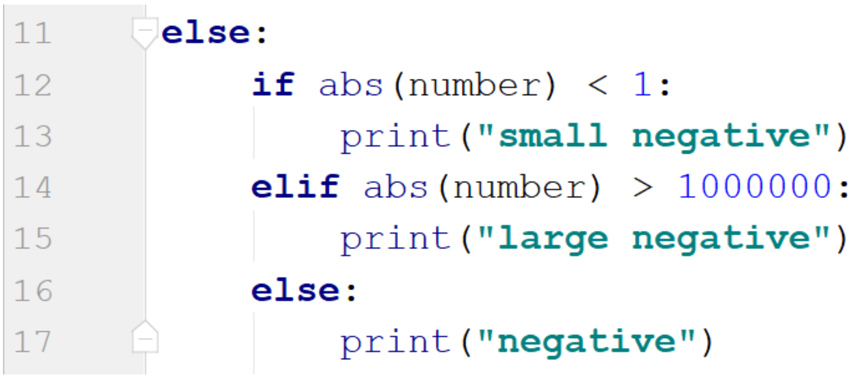
```python
if number == 0:
    print("zero")
```

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



```python
elif number > 0:
    if number < 1:
        print("small positive")
    elif number > 1000000:
        print("large positive")
    else:
        print("positive")
```

Then, we check if the number is negative. To check if the number is small or large, we use the absolute value of the negative number:



```python
else:
    if abs(number) < 1:
        print("small negative")
    elif abs(number) > 1000000:
        print("large negative")
    else:
        print("negative")
```

3. Word Reverse

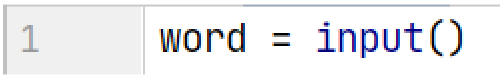
Write a program that receives a single word from a user, reverses it and prints it.

Example

Input	Output
Python	nohtyP
banana	ananab

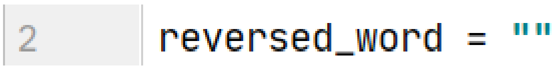
Hints

We read the word from the console:



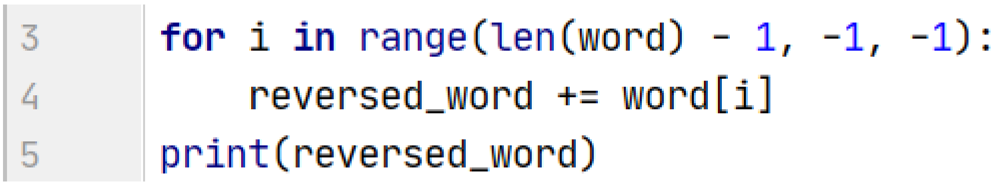
```
word = input()
```

Next, we create new variable to store the reversed word:



```
reversed_word = ""
```

After that, we create a for loop which would iterate backwards through the word and add each character to the new word:



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

- The starting point value of the loop should be the length of the word minus one position.
- We should loop until we reach the 0 index, so the stop value of the loop should be -1.
- Finally, the step is backwards, so the step size is -1.

4. Number Between 1 and 100

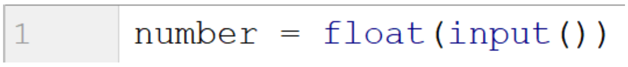
Write a program which reads different floating-point numbers from the console. When it receives a number between 1 and 100 inclusive, the program should stop reading and should print **"The number {number} is between 1 and 100"**.

Example

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

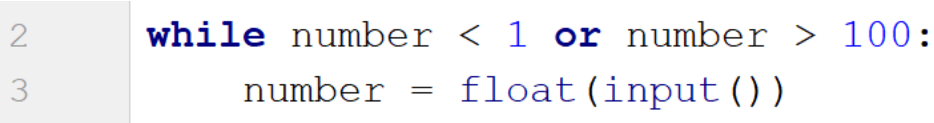
Hints

We start by reading a floating-point number:



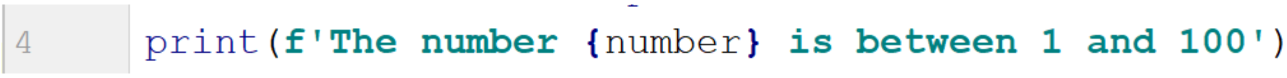
```
number = float(input())
```

Then, we create a while loop with a condition which checks if the given number is less than 1 or it is greater than 100. If the condition is met, the program should enter the body of the loop:



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

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



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

5. Patterns

Write a program which receives a number and creates the following pattern. The number represents the largest count of stars on one row.

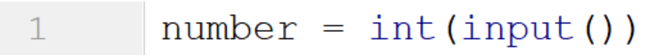
```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

Example

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

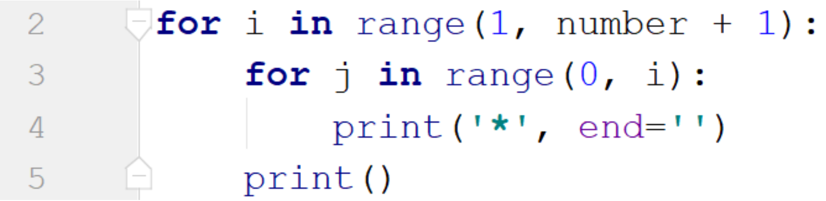
Hints

First, we read the number from the console:



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

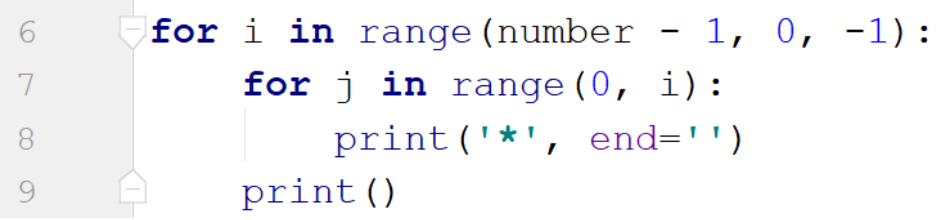
We create the first loop, which will print the half of the pattern, until **`i == number`**:



```python
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 numbers from **`0`** to **`i`** (**`i`** is not inclusive). We use **`end=''`** to stay on the same line. We print the new line after we draw all the stars for the particular line.

Next, we create the second loop which loop backwards as we need to decrease the number of stars with each line. It should draw one line less than the previous one:



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