# Lab: Basic Syntax, Conditional Statements and Loops

Problems for in-class lab for the [Python Fundamentals Course @SoftUni](https://softuni.bg/trainings/3609/programming-fundamentals-with-python-january-2022).

Submit your solutions in the SoftUni judge system at <https://judge.softuni.org/Contests/1718>.

## Largest of Three Numbers

Write a program that receives **three whole numbers** and prints 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:



Then we compare them and print the largest one:



## Number Definer

Write a program that reads a floating-point number and:

* prints **"zero"** if the number is zero.
* prints **"positive"** or **"negative"**.
* adds **"small"** if the absolute value of the number is less than 1 and it is not 0, or **"large"** if it exceeds

1 000 000.

### Example

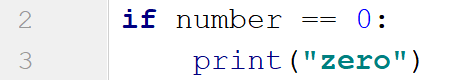
|  |  |
| --- | --- |
| **Input** | **Output** |
| 25 | positive |
| 0.7 | small positive |
| 435247392.921 | large positive |
| -0.005 | small negative |
| -103.21 | negative |
| -358583355123.001 | large negative |

### Hints

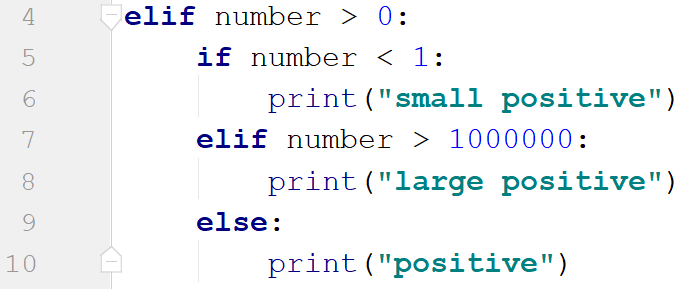
First, we read the number from the console as a float, because we are going to receive floating-point numbers:



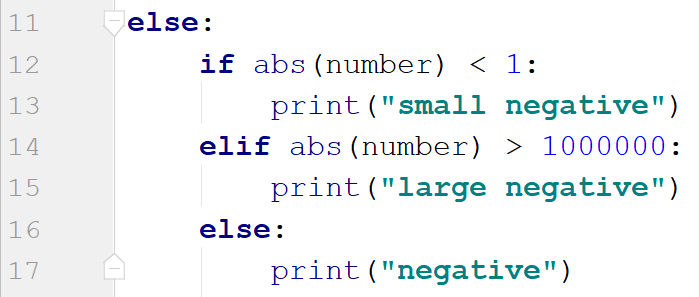
Then, we write a condition to check if the number is zero:



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



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:



## Word Reverse

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

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| Python | nohtyP |
| banana | ananab |

### Hints

We read the word from the console:



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



After that, we create a for loop which would iterate backward through the word and add each character to the new 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 backward, so the step size is -1.

## Number Between 1 and 100

Write a program that 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 |
| 0.5  90  -4  101 | The number 90.0 is between 1 and 100 |

### Hints

We start by reading a floating-point number:



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



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



## Patterns

Write a program that 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:



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



We create the inner 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 backward as we need to decrease the number of stars with each line. It should draw one line less than the previous one:

