# Lab: Functions Advanced

Problems for in-class lab for the [Python Advanced Course @SoftUni](https://softuni.bg/courses/python-advanced). Submit your solutions in the SoftUni judge system at <https://judge.softuni.bg/Contests/1838>

## Absolute Values

Write a program that receives a list of numbers and prints their **absolute value**. Use **abs()**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2.5 -3 -4.5 | [1.0, 2.5, 3.0, 4.5] |

## Rounding

Write a program that rounds all the given numbers. Use **round()**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1.0 2.5 3.0 4.5 | [1, 2, 3, 4] |

## Even Numbers

Write a program that receives a list of numbers. Print **only the even numbers** from the list. Use **filter()**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 2 3 4 | [2, 4] |

## Sort

Write a program that prints a **sorted** list of numbers in **ascending order**. Use **sorted()**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6 2 4 | [2, 4, 6] |

## Min Max and Sum

Write a program that prints **the min and max values** from a list and **the sum** of all the numbers in the list. Use **min(), max()** and **sum()**

### Example

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 4 6 | The minimum number is 2  The maximum number is 6  The sum number is: 12 |

## Multiplication Function

Write a function **called multiply** that can receive any amount of numbers as different parameters and returns the result of the multiplication of all of them. Submit **only your function** in judge.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| print(multiply(1, 4, 5))  print(multiply(4, 5, 6, 1, 3))  print(multiply(2, 0, 1000, 5000)) | 20  360  0 |

## Operate

Write a function **called** operate that receives an operator as the first argument and multiple numbers as additional arguments (\*args). The function should return the result of the operator applied to all the numbers. For more clarification, see the examples below. Submit **only the function** in the judge system.

### Examples

|  |  |  |
| --- | --- | --- |
| **Test Code** | **Output** | **Comment** |
| print(operate("+", 1, 2, 3)) | 6 | 1 + 2 + 3 = 6 |
| print(operate("\*", 3, 4)) | 12 | 3 \* 4 = 12 |

## Concatenate

Write a function called concatenate() that receives some strings, concatenates them and returns the result

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| print(concatenate("Soft", "Uni", "Is", "Great", "!")) | SoftUniIsGreat! |

## Person Info

Write a function **called** get\_info that receives a **name**, **age** and **town**, and returns a string in the format:   
**"This is {name} from {town} and he is {age} years old"**. Use dictionary unpacking when testing your function. Submit **only the function** in the judge system.

### Examples

|  |  |
| --- | --- |
| **Test Code** | **Output** |
| print(get\_info(\*\*{"name": "George", "town": "Sofia", "age": 20})) | This is George from Sofia and he is 20 years old |

## Character Combinations

Write a program that reads a **single** **string** and prints **all the possible combinations** of the **characters** in that string. Submit your solution in the judge system.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| abc | abc  acb  bac  bca  cba  cab |

## Chairs

Write a program that receives **names** on the first line (separated by comma and space **", "**) and number of **chairs** on the second line (an **integer**). Find all the ways to fit those people on the chairs. Print each combination on a separate line.

***Note: In the example below, "Peter, George" is same as "George, Peter", so we only print the first combination***

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Peter, George, Amy  2 | Peter, George  Peter, Amy  George, Amy |