# Exercise: Functions

Problems for exercise and homework for the [Python Fundamentals Course @SoftUni](https://softuni.bg/trainings/3450/programming-fundamentals-with-python-september-2021).   
Submit your solutions in the SoftUni judge system at <https://judge.softuni.org/Contests/1728>.

## Smallest of Three Numbers

Write a function that receives **three integer** numbers and **returns** the **smallest**. Print the result on the console. Use an appropriate name for the function.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  5  3 | 2 |
| 600  342  123 | 123 |
| 25  21  4 | 4 |

## Add and Subtract

You will receive **three** **integer numbers.**

Write functions named:

* sum\_numbers() that **returns** the **sum** of the **first two** integers
* subtract()that **returns** the **difference** between the **returned result** of the first function and the **third** integer

Wrap the two functions in a function named **add\_and\_subtract()** which will receive the three numbers as parameters. Print the result of the subtract() function on the console.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 23  6  10 | 19 |
| 1  17  30 | -12 |
| 42  58  100 | 0 |

## Characters in Range

Write a function that receives **two characters** and **returns** a **single string** **with all the characters in** **between them** (according to the **ASCII** code), separated by a single **space**. Print the result on the console.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| a  d | b c |
| #  : | $ % & ' ( ) \* + , - . / 0 1 2 3 4 5 6 7 8 9 |
| #  C | $ % & ' ( ) \* + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B |

## Odd and Even Sum

You will receive a **single number.** You should write a function that returns the **sum** of **all even** and **all odd** **digits** in a given number. The result should be returned as a single string in the format:

**"Odd sum = {sum\_of\_odd\_digits}, Even sum = {sum\_of\_even\_digits}"**

Print the result of the function on the console.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1000435 | Odd sum = 9, Even sum = 4 |
| 3495892137259234 | Odd sum = 54, Even sum = 22 |

## Even Numbers

Write a program that receives a sequence of numbers (integers) separated by a single space. It should print a list of **only the even numbers**. Use **filter()**.

### Example

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

## Sort

Write a program that receives a sequence of numbers (integers) separated by a single space. It should print 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 receives a sequence of numbers (integers) separated by a single space. It should print **the min and max values** of the given numbers and **the sum** of all the numbers in the list. Use **min(), max()** and **sum().**

The output should be as follows:

* On the first line: **"The minimum number is {minimum number}"**
* On the second line: **"The maximum number is {maximum number}"**
* On the third line: **"The sum number is {sum of all numbers}"**

### Example

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

## Palindrome Integers

A palindrome is a number that reads the same **backward as forward**, such as 323 or 1001. Write a function that receives a **list of positive integers**, separated by comma and space **", "**. The function should **check** if **each integer** is a **palindrome** - True or False. Print the result.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 123, 323, 421, 121 | False  True  False  True |  | 32, 2, 232, 1010 | False  True  True  False |

### Hints

* You can read more about palindromes here: <https://en.wikipedia.org/wiki/Palindrome>

## Password Validator

Write a function that checks if a given password is valid. Password validations are:

* It should be **6 - 10** (inclusive) characters long
* It should consist **only of** **letters** and **digits**
* It should have **at least 2** digits

If a password is **valid,** print **"Password** **is** **valid"**.

Otherwise, for every unfulfilled rule, print a message:

* **"Password must be between 6 and 10 characters"**
* **"Password must consist only of letters and digits"**
* **"Password must have at least 2 digits"**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| logIn | Password must be between 6 and 10 characters  Password must have at least 2 digits |
| MyPass123 | Password is valid |
| Pa$s$s | Password must consist only of letters and digits  Password must have at least 2 digits |

## Perfect Number

A perfect number is a **positive** integer that is equal to the **sum** ofits **proper positive divisors**. That is the sum of its positive **divisors,** excluding the number itself (also known as its **aliquot sum**).

Write a function that receives an integer **number** and **returns** **one** of the following messages:

* **"We have a perfect number!"** - if the number is **perfect**.
* **"It's not so perfect."** - if the number is **NOT** **perfect**.

Print the result on the console.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 6 | We have a perfect number! | 1 + 2 + 3 |
| 28 | We have a perfect number! | 1 + 2 + 4 + 7 + 14 |
| 1236498 | It's not so perfect. |  |

### Hint

Every perfect number is **half the sum** of all its positive divisors (including itself) => the sum of all positive divisors (all of which are divided without remainder) of 6 is 1 + 2 + 3 + 6 = 12. Half of 12 is 6 => 6 is perfect number.

* You could read more about the perfect number here: <https://en.wikipedia.org/wiki/Perfect_number>

## \* Loading Bar

You will receive a **single integer number** between **0** and **100** (inclusive) divisible by 10 without remainder (0, 10, 20, 30...). Your task is to create a function that returns a **loading bar** depending on the number you have received in the input. Print the result on the console. For more clarification, see the examples below.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 30 | 30% [%%%.......]  Still loading... |
| 50 | 50% [%%%%%.....]  Still loading... |
| 100 | 100% Complete!  [%%%%%%%%%%] |

## \* Factorial Division

Write a function that receives **two** integer numbers. Calculate the **factorial** of each number.

Divide the first result by the second and **print the division** formatted to the **second decimal** point.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 5  2 | 60.00 |  | 6  2 | 360.00 |

### Hints

* Read more about factorial here: <https://en.wikipedia.org/wiki/Factorial>