# Exercise: Writing and Calling Methods

Problems for exercise and homework for the ["Programming Fundamentals and Unit Testing" course @ SoftUni](https://softuni.bg/trainings/4256/programming-fundamentals-and-unit-testing-september-2023)  
You can check your solutions in [Judge](https://judge.softuni.org/Contests/4349/Methods-Writing-and-Calling-Methods-Exercise)

## Smallest of Three Numbers

Create a method that **prints out the smallest of three integer numbers**.

### Examples

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

## Vowels Count

Create a method that receives a **single string** and **prints out the number of vowels** contained in it.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| SoftUni | 3 |
| Cats | 1 |
| JS | 0 |

## Characters in Range

Create a method that receives **two characters** and prints all the **characters between them according to ASCII** (on a single line).

**NOTE:** If the second letter's ASCII value is less than that of the first one, then the two initial letters should be swapped.

### 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 |

## Password Validator

Create a program that checks if a given password is **valid**.

The password validation **rules** are:

* It should contain **6 – 10 characters (inclusive)**
* It should contain **only letters and digits**
* It should contain **at least 2 digits**

If it is **not valid**, for any unfulfilled rule **print the corresponding 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 |

### Hints

Write a method for each rule.

## Add and Subtract

### You will receive 3 integers. Create a method that returns the sum of the first two integers and another method that subtracts the third integer from the result of the sum method.

### Examples

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

## Middle Characters

You will receive a single string. Create a method that **prints the character found at its middle**. If the **length** of the string is **even**, there are **two middle characters**.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| aString | r |
| someText | eT |
| 3245 | 24 |

## NxN Matrix

Create a method that receives a single integer **n** and prints an **NxN** matrix using this number as a filler.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 | 3 3 3  3 3 3  3 3 3 |
| 7 | 7 7 7 7 7 7 7  7 7 7 7 7 7 7  7 7 7 7 7 7 7  7 7 7 7 7 7 7  7 7 7 7 7 7 7  7 7 7 7 7 7 7  7 7 7 7 7 7 7 |
| 2 | 2 2  2 2 |

## Factorial Division

Read **two integers**. Calculate the [factorial](https://en.wikipedia.org/wiki/Factorial) of each number. **Divide the first result by the second** and print the result of the division **formatted to the second decimal point**.

### Examples

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

## Palindrome Integers

Create a program that reads positive integers **until you receive** the "**END**" command. For each number, **print whether the number is a palindrome or not**. A palindrome is a number that reads the same backward as forward, such as 323 or 1001.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 123  323  421  121  END | false  true  false  true |  | 32  2  232  1010  END | false  true  true  false |

## Top Number

A **top number** is an integer that holds the following properties:

* Its sum of digits is divisible by 8, e.g. 8, 17, 88
* Holds at least one odd digit, e.g. 232, 707, 87578
* Some examples of top numbers are: 17, 161, 251, 4310, 123200

Create a program to print all top numbers in the range [1…n].

You will receive a single integer from the console, representing the end value.

### Examples

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |
| 50 | 17  35 |  | 100 | 17  35  53  71  79  97 |