Laboratory 2

Objective

The topic of this laboratory focuses mainly on lists, due to their importance in Python. Most of the list processing operation are involved in solving the proposed problems.

Problem 1

Given the text below (from Richard the Third of William Shakespeare), create a program with two functions: one counting the total number of words in this text and one counting all the vowels. Display the results at `stdout`.

"Now is the winter of our discontent
Made glorious summer by this sun of York;
And all the clouds that lour'd upon our house
In the deep bosom of the ocean buried."

Problem 2

Given a list of objects (strings or numbers) write a function that reverts the content of the list and print it to the stdout. Note: do not use the revert() method of Python lists.

Example:

```
input:  ['Monty', 'Python', 'and', 'the', 'Holy', 'Grail']
output: ['Grail', 'Holy', 'the', 'and', 'Python', 'Monty']
```

Problem 3

Ask user for a string and determine in that string is a **palindrome** or not. A palindrome is a string that reads the same forwards and backwards. (e.g. kayak, level, minim, radar, rotator, ...)

Problem 4

Take two lists and write a program that returns a list containing only the common elements between the lists. Use a function for determining the common elements.

Example:

```
[1, 2, 3, 4, 5, 6, 7, 8]
[2, 4, 9, 11, 33]
result: [2, 4]
```

Problem 5

Write a function that will check if a number between 1000 and 2000 is divisible with 7 but is not divisible with 5, then create a list with all these numbers. Display this list at `stdout`.

Problem 6

Write a program using a function that computes the factorial of a number. The *factorial* of a number *n* is symbolised as *n!* and it is the product of all positive integers less than or equal to *n*. For example,

$$5!=5 \times 4 \times 3 \times 2 \times 1=120$$

The value of 0! is 1, according to the convention for an empty product.

Problem 7

Given an array of ints, do the following processing:

1) Remove duplicates
2) Remove the minimum and the maximum element from the remaining array
3) Compute the mean average (integer) of the remaining elements.
4) Display the result

Input array:

```
[10, 20, 20, 30, 30, 56, 67, 75, 22, 10, 33]
```