

ALP

EXERCISE SHEET
ACADEMIC ACTIVITY

Algorithms and Data Structures

CURRICULAR UNIT

Sheet 06 – Lists

1. Create the **aboveAverage** function that receives a list of 10 integers (numbers entered by the user) and returns how many of these numbers are above the average.
2. Create a **generateNumbers** function that allows you to generate a random Euromillions key: 5 numbers between 1 and 50, as well as the stars (two stars between 1 and 12).

The generateNumbers function must have 3 input arguments: lower limit, upper limit, and number of numbers to generate: **generateNumbers(lowerLimit, upperLimit, numbers)**.

For example, to generate a random Euromillions key, you must invoke the function with the following arguments:

generateNumbers(1,50, 5) -> generates 5 numbers between 1 and 50

generateNumbers(1,12,2) -> generates 2 numbers between 1 and 12 (stars)

The function must return a list with the generated numbers. Obviously, the function cannot return repeated numbers or repeated stars!

At the end, print the generated Euromillions key and ask the user if he wants to generate a new key (Y/N).

Example:

C:\WINDOWS\py.exe

Chave do Euromilhões: [26, 36, 11, 27, 47] Estrelas: [3, 6]

Deseja gerar nova Chave(S/N)? ☐

3. Create a program that reads the scores of 10 participants in a programming contest (the scores must be validated between 0 and 20, using a try-except structure).

The program must invoke a function, **positiveList**, that receives the list of 10 scores and returns a new list with only the positive scores (≥ 10).

Example:

```
C:\WINDOWS\py.exe
Pontuação: 12
Pontuação: 9
Pontuação: 6
Pontuação: 7
Pontuação: 16
Pontuação: 17
Pontuação: 14
Pontuação: 2
Pontuação: 3
Pontuação: 1
Pontuações positivas: [12, 16, 17, 14]
```

4. Change the previous program to also include the input of the names of the contest participants.

The **positiveList** function should now return two lists: one with scores and another with names of participants who obtained positive scores (≥ 10).

Example:

```
C:\WINDOWS\py.exe
Participantes com pontuações positivas
Nomes      : ['António', 'Carla', 'Maria']
Pontuações: [12, 15, 18]
```

5. Create a program that allows you to read the sales volume over the 12 months of the year (from January to December).

The program must include the calling of **3 functions** which return, respectively:

a) the month with the highest volume sales

b) the month with the lowest sales

c) the average sales value

```
C:\WINDOWS\py.exe
faturação do mês Janeiro : 800
faturação do mês Fevereiro : 900
faturação do mês Março : 1100
faturação do mês Abril : 1210
faturação do mês Maio : 1320
faturação do mês Junho : 1570
faturação do mês Julho : 1400
faturação do mês Agosto : 1300
faturação do mês Setembro : 750
faturação do mês Outubro : 900
faturação do mês Novembro : 1400
faturação do mês Dezembro : 1350

Mês de maior faturação : Junho
Mês de menor faturação : Setembro
Valor médio de faturação 1166.67:
```

6. Write a program that reads a list of 10 integers.

Then, given a search value, invoke the function **searchNumber**(list, searchValue) which should return the positions where the search value is found, in the list.

If the search value does not exist in the list, an appropriate message should appear.

7. The Meteorology Institute intends to record the total amount of rainfall that occurred in each month, over the course of a year (from January to December).

Implement the rainfall function that receives the list of rainfall for each month (by input), and prints this same data (rainfall list), but sorted in descending order of rainfall.

Upgrade to version 2.0: also print the names of the respective months, as in the image!

```
C:\WINDOWS\py.exe
Pluviosidade no mês de Janeiro : 150
Pluviosidade no mês de Fevereiro : 190
Pluviosidade no mês de Março : 225
Pluviosidade no mês de Abril : 270
Pluviosidade no mês de Maio : 145
Pluviosidade no mês de Junho : 110
Pluviosidade no mês de Julho : 90
Pluviosidade no mês de Agosto : 60
Pluviosidade no mês de Setembro : 80
Pluviosidade no mês de Outubro : 120
Pluviosidade no mês de Novembro : 175
Pluviosidade no mês de Dezembro : 185

Abril 270
Março 225
Fevereiro 190
Dezembro 185
Novembro 175
Janeiro 150
Maio 145
Outubro 120
Junho 110
Julho 90
Setembro 80
Agosto 60
```

8. Given a list of N elements (N must be requested from the user), create a function that orders the list and allows you to generate another list without duplicate values.

Example initial list:

2	4	6	6	10	12	12	16	18	20
---	---	---	---	----	----	----	----	----	----

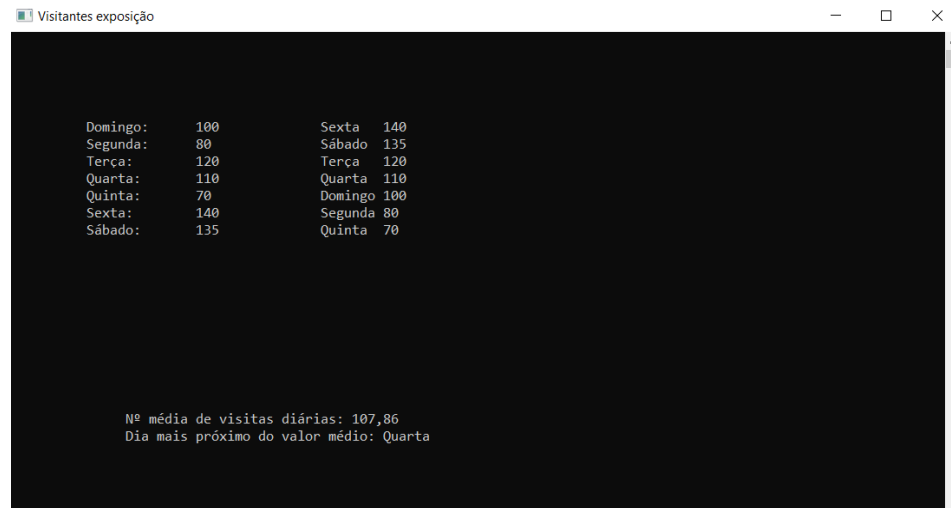
Generated list:

2	4	6	10	12	16	18	20
---	---	---	----	----	----	----	----

9. Implement a program that allows you to read the number of visitors to an exhibition, which runs from Sunday to Saturday.

Next, create a function that allows you to list the number of daily visitors in descending order, as in the image below.

Also indicate, at the end, the average number of visitors per day (to 2 decimal places) and the day that came closest to the average number of visitors.



```

Visitantes exposição

Domingo: 100      Sexta 140
Segunda: 80       Sábado 135
Terça: 120        Terça 120
Quarta: 110       Quarta 110
Quinta: 70        Domingo 100
Sexta: 140        Segunda 80
Sábado: 135       Quinta 70

Nº média de visitas diárias: 107,86
Dia mais próximo do valor médio: Quarta
  
```