jgromero@inf.uc3m.es

Lesson 7. Search, Sort and Merge Algorithms
Exercises

Exercise 1. Write a C program to find the number of times that each digit is repeated in an integer number *n*. The integer number must be read from the keyboard. The counting of repeated digits must be printed on the screen in decreasing order of repetitions. For example:

Input *n*:
```
2254565
```

Output:
```
5 is repeated 3 times
2 is repeated 2 times
4 is repeated 1 times
6 is repeated 1 times
```

Exercise 2. Write a C program to calculate the distribution of seats in the Parliament by applying the D'Hont Law.

Ten political parties stand in the general elections for the electoral district of Madrid (34 seats). The results of the voting, which is the input of the program, are stored in an array –the votes obtained by the party *i* are stored in the position *i* of the array.

The program must print on the screen an ordered list with the parties, the votes, and the seats –in decreasing order of seats–, according to the following rules:

Simplified D'Hont Law
A list with the parties and the number of votes obtained –sorted in decreasing order—is created. From this list, a table is created:

- The rows of the table represent the parties
- The columns of the table stores the result of the division of the number of votes of the parties by 1, 2, 3, etc. until the number of seats of the district

The seats are assigned to the parties with the largest values in their corresponding columns.

For example, for a 6-seat district, given the following data:

Party	Votes	Votes / 2	Votes / 3	Votes / 4	Votes / 5	Votes / 6
A	554.724	277.362	184.908	138.681	110.945	92.454
B	400.328	200.164	133.443	100.082	80.066	66.721
C	356.810	178.405	118.937	89.202	71.362	59.468
D	122.300	61.150	40.767	30.575	24.460	20.383

(Green cells represent the input data; blue cells represent the assignment table.)

The largest values of the table are:

1º seat Value 554.724 Party A
2º seat Value 400.328 Party B
3º seat Value 356.810 Party C
4º seat Value 277.362 Party A
5º seat Value 200.164 Party B
6º seat Value 184.908 Party A

Therefore, the program must print on the screen:

```
Party A: 3 seats
Party B: 2 seats
Party C: 1 seats
Party D: 0 seats
```

Exercise 3. Write a C program that reads 20 positive integer values from the keyboard and calculates the sum of the introduced even numbers, the sum of the introduced odd numbers, and the mean of the introduced odd numbers. Sort the numbers in decreasing order with *Bubblesort*.

Extend the exercise to let the user select the sorting method: *Bubblesort, Selectionsort, Insertionsort.*

Exercise 4. Write a C program to calculate the score of the drivers of the F3000 championship.

The championship has 25 races (*N*) and 5 drivers (*D*). The score gained by a driver is assigned according to the table below:

Position	1	2	3	4	5
Score	50	20	10	5	0

The program must read from the keyboard, for each race, the position of each driver; i.e., the program must ask to the user:

```
Results of race 1
1st: 2
2nd: 3
3rd: 5
4th: 1
5th: 4

Results of race 2
1st: 3
2nd: 5
3rd: 2
…
```

(**NOTE**: Use arrays (1-dimension or 2-dimension) to represent the results of the championship. The drivers are named from 1 to 5.)

The output of the program must be the final score of the three best drivers in decreasing order.

For example, given the following data:

	1st	2th	3rd	4th	5th
Race 1	2	3	5	1	4
Race 2	3	5	2	4	1
Race 3	5	1	4	2	3

The results are:

Driver	1	2	3	4	5
Score	25	65	70	15	80

And the final output of the program is:

```
Winner: Driver 5
Second: Driver 3
Third: Driver 2
```