

PROBLEM – Lesson 5: The Voting

Statement

Students of the last course of a UAB degree aim to decide the destination of the end-of-year trip by voting. There are 4 options: 1-London, 2-Paris, 3-Rome and 4-Ibiza. The voting will be managed through a computer program that must be implemented.

- There are 80 students, and each student has a university identification number (NIU).
- Students will vote non-consecutively (i.e. they will not follow the order of their NIU).
- The vote will be public, which means that several lists must be maintained:
 - the NIU of students who have already voted
 - the votes cast
- Voting is mandatory, all students must vote.
- At the end of the voting, the counting will be done, and the voting results will be shown.

Exercise 1

Make a function called *ReadNumber*, which receives two integer values (minimum and maximum), where $\text{minimum} < \text{maximum}$. The function will read a number via keyboard and will check that it belongs to the interval $[\text{minimum}, \text{maximum}]$ (closed interval).

The function will inform the user with the following message:

Enter value between XX and YY:

where XX and YY will be the minimum and maximum values passed as parameters.
In case of error, the message will be:

Error: Incorrect value

and will ask for the value again. This process will be repeated until a correct value is entered.
If everything is correct, the function will return the value read (within the range).

Exercise 2

Do a procedure called *Vote* that records the destination chosen by a student.
The procedure will receive a list of all students who have voted and a list of votes cast.

The user will be asked to enter the NIU and the program will check if the student has already voted. If so, the following message will be printed:

Error: You have already voted

If the student has not voted yet, the program will ask the user to enter the vote using the function *ReadNumber* from the previous exercise. Remember that votes are coded with values between 1 and 4 (both inclusive). Once the value is correct it will be added to the list of votes cast and the student's NIU will be added to the list of those who have voted.

Exercise 3

Implement a function called *Scrutiny* that counts the votes. To do this we will pass, as a parameter, the list of votes cast.

The function will return a list with the vote count for each option. The list will contain the votes of each option according to the order of coding: 1-London, 2-Paris, 3-Rome and 4-Ibiza.

Exercise 4

Make a function called *MaxIndex*. This function will receive a list of integers and it will return the index of the position of the maximum value.

Exercise 5

Implement an algorithm (the main program) that follows the following steps:

1. Declarations and initializations.
2. Call the procedure in Exercise 2, *Vote*, for all students in the course to vote.
3. Call the procedure of Exercise 3, *Scrutiny* to know the results of the vote.
4. Call the procedure in Exercise 4, *MaxIndex* to find out the most voted option.
5. Print the message:

The chosen destination is <destination> with <votes> votes

where <destination> is the most voted destination (London, Paris, Rome, or Ibiza) and <votes> is the number of votes obtained by this destination.