

Exercise paper#5

Stefano Genetti
stefano.genetti@studenti.unitn.it

April 8, 2022

1 Exercise 1

Predict the output of the following (meaningless) code.

```
7  *****/
8
9  #include <iostream>
10
11 #define MAX_ARRAY_LENGTH 10
12
13 using namespace std;
14
15 int main()
16 {
17     int uno[MAX_ARRAY_LENGTH] = {2, 3, 1, 10, -1, 3, 2, 4, 1, 67};
18     int due[MAX_ARRAY_LENGTH];
19
20     int m = MAX_ARRAY_LENGTH/2;
21
22     for(int i=m; i>=0; i--){
23         due[i] = uno[i]*2;
24     }
25
26     for(int i=m+1; i<MAX_ARRAY_LENGTH; i++){
27         if(uno[i]%2==0){
28             due[i]=uno[i];
29         }else{
30             due[i]=2;
31         }
32     }
33
34     cout<<"OUTPUT"<<endl;
35     for(int j=0; j<MAX_ARRAY_LENGTH; j++){
36         cout<<"uno["<<j<<"] = "<<uno[j];
37         cout<<" - due["<<j<<"] = "<<due[j]<<endl;
38     }
39
40     return 0;
41 }
42
```

2 Exercise 2

Given a matrix of integers, write a C++ program which prints the central cross of numbers. Look at the example to better understand the expected behaviour:

Example:

Input					Output				
1	2	2	3	1	1	2	2	3	1
9	2	6	4	5	9	2	6	4	5
9	1	7	5	1	9	1	7	5	1
5	6	1	9	1	5	6	1	9	1
5	6	7	17	6	5	6	7	17	6

In this case on the console you have to print something like:

$\{2 - 6 - 7 - 1 - 7\} - \{9 - 1 - 7 - 5 - 1\}$


Input Assumptions:

- The matrix is a square matrix (https://it.wikipedia.org/wiki/Matrice_quadrata)
- The size of the input matrix is always an odd number. In this way it is always possible to find a center.

3 Exercise 3

Declare a matrix of integers with global scope, so outside any functions like this:

```
7  *****/
8
9  #include <iostream>
10
11 using namespace std;
12
13
14 /**!!!LOOK HERE!!!*/
15
16 int myMatrix[5][5] = { 12, 23, 12, 3, 2,
17                        32, 21, 33, 3, 6,
18                        11, 25, 55, 2, 7,
19                        12, 23, 99, 7, 8,
20                        44, 55, 88, 6, 9
21 };
22
23 /**!!!!!!!!!!!!!!*/
24
25 int main()
26 {
27     for(int i=0; i<5; i++){
28         for(int j=0; j<5; j++){
29             cout<<myMatrix[i][j]<<" - ";
30         }
31         cout<<endl;
32     }
33     return 0;
34 }
35
```



At this point write a function `max_matrix()` that, given the global matrix `myMatrix` returns the largest integer which populates the matrix.

Example:

Using the matrix `myMatrix` declared in the previous image, the expected behaviour is as follows:

Istruzione: `cout<<"Il valore maggiore della matrice: "<<max_matrix()<<endl`

Output: Il valore maggiore della matrice: 99

4 Exercise 4

Write a function `vowel_match()` which takes as input arguments two strings `s1` and `s2`. The function compares the two strings and returns `true` if the two strings have the same number of vowel letters and `false` otherwise.

Example:

```
int main()
{
    string stringa1 = "ANNA";
    string stringa2 = "PENNA";

    if(vowel_match(stringa1, stringa2)==true){
        cout<<stringa1<<" and "<<stringa2<<" contain the same number of vowels"<<endl;
    }else{
        cout<<stringa1<<" and "<<stringa2<<" do not contain the same number of vowels"<<endl;
    }

    //OUTPUT: ANNA and PENNA contain the same number of vowels

    /*=====*/

    string stringa3 = "ROMA";
    string stringa4 = "AIUOLA";

    if(vowel_match(stringa3, stringa4)==true){
        cout<<stringa3<<" and "<<stringa4<<" contain the same number of vowels"<<endl;
    }else{
        cout<<stringa3<<" and "<<stringa4<<" do not contain the same number of vowels"<<endl;
    }

    //OUTPUT: ROMA and AIUOLA do not contain the same number of vowels

    return 0;
}
```