

[Home](#)▼ [ACSO](#)[Alte probleme](#)[Laborator 1 + 2](#)[Laborator 3](#)[Laborator 4](#)[Laborator 5](#)**[Laborator 6](#)**[Seminar 1](#)[Seminar 2](#)[Seminar 3](#)[Seminar 4](#)[Seminar 5](#)[Seminar 6](#)[Seminar 7](#)[Sitemap](#)[ACSO](#) >

## Laborator 6

Rezolvati exercitiile de mai jos, sau cele din laboratoarele precedente nerezolvate.

```
1.
#include "stdafx.h"

#include <stdio.h>
#include <malloc.h>
#include <string.h>

unsigned nrBits( int ){
    _asm{
        //returneaza numarul de biti setati in reprezentarea interna a numarului primit ca parametru
    }
}

unsigned isPrime( int ){
    _asm{
        //returneaza 1 daca numarul primit ca parametru este prim, 0 in caz contrar
    }
}

int palindrom( unsigned int ){
    _asm{
        //returneaza 1 daca numarul primit ca parametru este palindrom, 0 in caz contrar
    }
}

unsigned short powerOf2( int , int* ){
    _asm{
        //Se da un vector prin intermediul a doi parametri: primul reprezinta numarul de elemente din
        //vector iar al II-lea este pointer catre primul element din vector.
        //Functia returneaza numarul de elemente din vector - puteri ale lui doi.
    }
}

int progresie_aritmetica(int, int*){
    _asm{
        //Se da un vector prin intermediul a doi parametri: primul reprezinta numarul de elemente din
        //vector iar al II-lea este pointer catre primul element din vector.
        //Functia returneaza 1 daca elementele vectorului formeaza o progresie aritmetica, 0 in caz
        //contrar
    }
}

int progresie_geometrica(int, int*){
    _asm{
        //Se da un vector prin intermediul a doi parametri: primul reprezinta numarul de elemente din
        //vector iar al II-lea este pointer catre primul element din vector.
        //Functia returneaza 1 daca elementele vectorului formeaza o progresie geometrica, 0 in caz
        //contrar
    }
}

int toUpper(char* ){
    _asm{
```

```

        //Faceti conversia sirului de caractere primit ca parametru, returnand numarul de caractere
        modificate.
    }
}

int nrTok( char* ){
    _asm{
        //Sa se returneze numarul de cuvinte din textul primit ca parametru. Separatorii sunt: ',', ' ',
        '!'
    }
}

```

2. Definiti o structura, numita Rezultat, ce contine doua campuri de tip int.

Scrieti doua functii ce primesc ca parametrii un numar L, C, o matrice M, si o referinta catre un obiect de tip Rezultat (unde L si C reprezinta numarul de linii si coloane din matrice). Una dintre functii va lucra cu matrici alocate dinamic, iar cealalta cu matrici alocate static. Functia trebuie sa populeze parametrul de tip Rezultat, punand in unul din campuri suma maxima pe linnii, iar in celalalt suma maxima pe coloane.

3. Scrieti o functie care primeste ca parametru o matrice de dimensiune 10x10, populata cu numere pozitive, si inca doi parametrii L si C, ce reprezinta o pozitie de inceput.

Functia returneaza 0, daca in matrice putem face un drum ciclic, sau 1 daca nu.

Drumul se va face respectand urmatoarele reguli:

1. Daca in pozitia curenta avem valoarea 0, ne oprim.
2. Daca in pozitia curenta avem valoare !=0, ne mutam in pozitia data de numarul de pe pozitia curenta, la linia data de cifra zecilor si coloana data de cifra unitatilor.

## Comments

You do not have permission to add comments.