Laborator 4

Teorie:

- Link

Probleme rezolvate:

1. Implementati o functie care primeste ca parametru doua numere si le interschimba valoarea

```
#include "stdafx.h"
#include<iostream>
using namespace std;
void interschimba(int *a, int *b) {
 _asm {
  mov eax, [ebp+8] //In eax am adresa lui a
  mov ebx, [ebp+12] //In ebx am adresa lui b
  mov ecx, [eax] //In ecx am valoarea de la adresa lui a, adica a
  mov edx, [ebx] //In edx am valoarea de la adresa lui b, adica b
  mov [eax], edx //La adresa lui a scriu valoarea lui b
  mov [ebx], ecx //La adresa lui b scriu vechea valoare a lui a.
 }
}

int main()
{
 int a, b;
 cout << "a=";
 cin >> a;
 cout << "b=";
 cin >> b;
 interschimba(&a, &b);
 cout << "a=" << a << " b=" << b << "\n";
    return 0;
}
```

2. Calculati suma elementelor pare dintr-un vector.

```
#include "stdafx.h"
#include<iostream>
using namespace std;

int suma_pare(int *, int) {
 _asm {
  mov eax, 0 //Aici facem suma
  mov esi, [ebp+8] //adresa primului elem din vector
  mov ecx, [ebp+12] //contor
 _start_while:
  mov ebx, [esi] //elementul curent din vector
  add esi, 4 //+4 pentru ca int are size 4
  test ebx,1 //daca numarul nu e impar, nu adunam
  jnz skip_add
  add eax, ebx
 skip_add:
  loop _start_while
  // in eax avem suma
 }
}

int main()
{
```

```
 int a, b;
 int v[5] = { 5,1,2,3,6 };
 int *p = v;
 int s;
 _asm {
  push 5
  push p
  call suma_pare
  mov s, eax
  add esp, 8
 }
 cout << "Suma:" << s << "\n";
    return 0;
}
```

3. Faceti suma a doua matrici patratice, declarate static.

```
#include "stdafx.h"
#include<iostream>
using namespace std;
void suma_matrice_statice(int *, int*, int ) {
 // A = A + B
 _asm {
  mov edi, [ebp + 8] //Destinatia va fi A
  mov esi, [ebp + 12]//Sursa va fi B
  mov ecx, [ebp + 16]//Counter.
  //In memorie matricea e ca un vector => putem face suma pozitie cu pozitie
  //de n^2 ori
  mov eax, [ebp+16]
  mov edx, 0
  mov ebx, [ebp+16]
  mul ebx
  mov ecx, eax //counter = n*n
 _while:
  mov ebx, [esi]
  add [edi], ebx
  add esi, 4
  add edi, 4
  loop _while
 }
}
int main()
{
 int n = 3;
 int A[3][3] = {
  {1,2,3},
  {2,3,5},
  {1,0,0}
 };
 int *pa = A[0];
 int B[3][3] = {
  {1,0,0},
  {0,1,0},
  {1,1,1}
 };
 int *pb = B[0];
 _asm {
  push n
  push pb
  push pa
  call suma_matrice_statice
  add esp, 12
 }
 for (int i = 0; i < n; ++i) {
  for (int j = 0; j < n; ++j) {
   cout << A[i][j] << ' ';
  }
  cout << '\n';
 }
```

```
    return 0;
}
```

4. Faceti suma a 2 matrici patratice, declarate dinamic.

```
#include "stdafx.h"
#include<iostream>
using namespace std;

void suma_matrice_dinamice(int *, int*, int) {
 _asm {
  mov edi, [ebp+8]
  mov esi, [ebp+12]
  mov ecx, [ebp+16]
  //De data asta facem cu doua foruri. Alocam memorie pe sitva pt i si j
  push 0
  push 0
 _start_for_i:
  mov ecx, [esp]
  cmp ecx, [ebp+16] //i<n?
  jge _end_for_i
  //in esi punem pointerul la linia curenta din B
  mov esi, [ebp+12]
  mov esi, [esi + 4*ecx]
  //la fel si in edi punem linia curenta din A
  mov edi, [ebp+8]
  mov edi, [edi + 4*ecx]
  _start_for_j:
   mov ecx, [esp+4]
   cmp ecx, [ebp+16] // j < n?
   jge _end_for_j
   mov ebx, [esi+4*ecx] //B[i][j]
   add [edi+4*ecx], ebx
   inc [esp+4] //++j
   jmp _start_for_j
  _end_for_j:
  inc [esp] //++i si o luam de la capat, j=0
  mov [esp+4], 0
  jmp _start_for_i
 _end_for_i:
  //gata. Dealocam de pe stiva
  add esp, 8
 }
}

int main()
{
 int n = 3;
 int **A;
 //Matricea va fi pointer de pointeri.
 A = (int**)malloc(n*sizeof(int*)); //Alocam memorie pentru cei n pointeri pentru lin
 for (int i = 0; i < n; ++i) {
  //alocam fiecare linie.
  A[i] = (int*)malloc(n*sizeof(int));
 }
 int **B;
 B = (int**)malloc(n*sizeof(int*));
 for (int i = 0; i < n; ++i) {
  B[i] = (int*)malloc(n*sizeof(int));
 }
 A[0][0] = 1; A[0][1] = 2; A[0][2] = 3;
 A[1][0] = 2; A[1][1] = 3; A[1][2] = 5;
 A[2][0] = 1; A[2][1] = 0; A[2][2] = 0;

 B[0][0] = 1; B[0][1] = 1; B[0][2] = 0;
 B[1][0] = 2; B[1][1] = 1; B[1][2] = 1;
 B[2][0] = 1; B[2][1] = 0; B[2][2] = 1;
 _asm {
  push n
```

```
 push B
 push A
 call suma_matrice_dinamice
 add esp, 12
}
for (int i = 0; i < n; ++i) {
 for (int j = 0; j < n; ++j) {
  cout << A[i][j] << ' ';
 }
 cout << '\n';
}
   return 0;
}
```

Probleme nerezolvate

1. Scrieti o functie ce primeste ca parametru un sir de caractere (terminat in 0x00) si returneaza numarul de vocale din acesta.
2. Scrieti o functie ce primeste ca parametru un vector int si lungimea acestuia si returneaza numarul de numere prime din el (Hint: Faceti o functie separata pentru testarea primalitatii si nu uitati sa salvati registrii din apelant)
3. Scrieti o functie ce primeste ca parametrii 2 vectori sortati imrepuna cu lungimile lor, si adresa unui al treilea vector. Functia va realiza interclasarea celor doi vectori, rezultatul fiind pus in al treilea.
4. Construiti matricea unitate, pentru o matrice declarata static si una declarata dinamic.

Comments

You do not have permission to add comments.