**Практическая работа № 10**

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил | Студент гр. МФ-11 | Лазаренко А. В. | 03.05.2021 |
| Проверил | Ст. преподаватель | Меняйлов Е. С. |  |

**Тема работы:**

«Организация циклов и работа с массивами»

**Цель работы:**

Ознакомиться с представлением многомерных массивов в памяти. Задав матрицу целочисленных данных (матрицу А), реализовать ее обработку, как указано в варианте, на языке Ассемблера.

**Задание:**

1. Проанализировать задание, составить алгоритм решения

поставленной задачи, с учетом применения команд ассемблера, определить

необходимые входные и выходные данные;

2. Написать на языке ассемблера процедуру, выполняющую

поставленную задачу;

3. Написать программу на С++, которая вводит значения входных

параметров и осуществляет проверку на аномалии при необходимости

выдает предупреждающее сообщения, после чего проводит обработку

массива на ассемблере и выводит результат на экран;

4. Протестировать программу на корректных и аномальных входных

данных.

**Вариант 11**

**Номер 1**

**Дан одномерный массив из 25 элементов. Необходимо заменить максимальным элементом все нулевые элементы массива.**

#define N 25

#include <iostream>

using namespace std;

int main(void)

{

long int result\_maximum = 0;

long int num\_array[N];

srand(time(0));

cout << "The array: ";

for (int i = 0; i < N; i++)

{

num\_array[i] = (rand() % 2 == 1) ? -(rand() % 100) : rand() % 100;

cout.width(3); cout << num\_array[i] << ' ';

}

num\_array[0] = num\_array[12] = num\_array[24] = 0;

\_\_asm

{

mov ecx, N

mov ebx, 4

xor eax, eax

mov edx, num\_array[eax]

m\_begin:

cmp edx, num\_array[ebx]

jge m\_greater\_equal

jl m\_lower

m\_greater\_equal:

mov eax, edx

jmp m\_exit

m\_lower:

mov edx, num\_array[ebx]

mov eax, edx

jmp m\_exit

m\_exit:

add ebx, 4

loop m\_begin

mov result\_maximum, eax

xor ebx, ebx

mov ecx, N

m\_start:

mov edx, num\_array[ebx]

cmp edx, 0

je m\_replace

add ebx, 4

loop m\_start

jmp m\_end

m\_replace:

mov num\_array[ebx], eax

add ebx, 4

loop m\_start

m\_end:

mov eax, 4C00h

}

cout << endl << "The found maximum is: " << result\_maximum << endl << "The changed array: ";

for(int i = 0; i < N; i++)

{ cout.width(3); cout << num\_array[i] << ' '; }

cout << endl;

return 0;

}

**Результаты выполнения программы**

1)



2)



3)



**Номер 2**

**Дан двумерный массив из 25 элементов – матрица 5 х 5. Найти сумму и произведение всех ее отрицательных элементов.**

#define N 5

#include <iostream>

using namespace std;

int main(void)

{

long int num\_array[N][N];

long int sum = 0, product = 0;

srand(time(0));

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

num\_array[i][j] = (rand() % 2 == 1) ? -(rand() % 10) : rand() % 10;

cout.width(3); cout << num\_array[i][j] << ' ';

}

cout << endl;

}

long int n\_size = N \* sizeof(long int);

\_\_asm

{

mov ecx, N

mov edx, 1

lea ebx, num\_array

m\_beg\_n:

push ecx

mov ecx, N

xor esi, esi

m\_beg\_m:

mov eax, [ebx][4 \* esi]

cmp eax, 0

jl m\_sum

jge m\_cycle

m\_sum:

add sum, eax

jmp m\_product

m\_product:

imul edx, eax

jmp m\_cycle

m\_cycle:

inc esi

loop m\_beg\_m

add ebx, n\_size

pop ecx

loop m\_beg\_n

m\_end:

mov product, edx

}

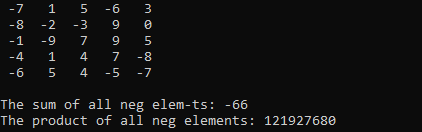
cout << endl << "The sum of all neg elem-ts: " << sum << endl << "The product of all neg elements: " << product << endl;

return 0;

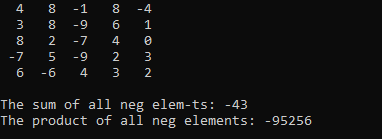
}

**Результаты выполнения программы**

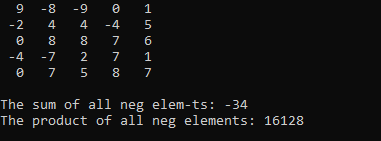
1)



2)



3)



**Задание\*:**

**Номер 1\***

#define NUMBER\_OF\_ELEMENTS 25

#include <iostream>

using namespace std;

int main(void)

{

float num\_array[NUMBER\_OF\_ELEMENTS];

float result\_maximum = 0.0f, temp = 0.0f;

srand(time(0));

cout << "The array: ";

for (int i = 0; i < NUMBER\_OF\_ELEMENTS; i++)

{

num\_array[i] = (rand() % 2 == 1) ? -(0.01 \* (rand() % 100) + rand() % 100) : 0.01 \* (rand() % 100) + rand() % 100;

}

num\_array[rand() % 25] = num\_array[rand() % 25] = num\_array[rand() % 25] = num\_array[rand() % 25] = num\_array[rand() % 25] = 0;

for (int i = 0; i < NUMBER\_OF\_ELEMENTS; i++)

cout << num\_array[i] << ' ';

\_\_asm

{

mov ecx, NUMBER\_OF\_ELEMENTS - 1

xor esi, esi

fld num\_array[esi]

add esi, 4

m\_begin:

fld num\_array[esi]

fxch st(1)

fcomi st, st(1)

jae m\_greater\_equal

jb m\_lower

m\_greater\_equal:

fst st(1)

jmp m\_exit

m\_lower:

fld st(1)

fstp temp

jmp m\_exit

m\_exit:

add esi, 4

fstp temp

loop m\_begin

fst result\_maximum

fninit

mov ecx, NUMBER\_OF\_ELEMENTS

xor esi, esi

m\_beg\_replace:

fld num\_array[esi]

fldz

fcomi st, st(1)

je m\_equal

jne m\_end

m\_equal:

fld result\_maximum

fstp num\_array[esi]

m\_end:

add esi, 4

fstp temp

fstp temp

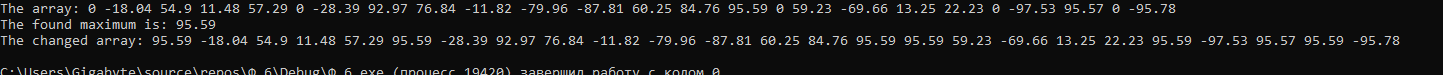
loop m\_beg\_replace

}

**Результаты выполнения программы**

1) 

2) 

3) 

**Номер 2\***

#define N 5

#include <iostream>

using namespace std;

int main(void)

{

float num\_array[N][N];

float sum = 0.0, product = 0.0, temp = 0.0;

srand(time(0));

for (int i = 0; i < N; i++)

{

for (int j = 0; j < N; j++)

{

num\_array[i][j] = (rand() % 2 == 1) ? -(0.01 \* (rand() % 100) + rand() % 4) : 0.01 \* (rand() % 100) + rand() % 4;

cout.width(5); cout.precision(4); cout << num\_array[i][j] << ' ';

}

cout << endl;

}

long int n\_size = N \* sizeof(long int);

\_\_asm

{

mov ecx, N

mov eax, 1

lea ebx, num\_array

fld1

fldz

m\_beg\_n:

push ecx

mov ecx, N

xor esi, esi

m\_beg\_m:

fldz

fld [ebx][esi \* 4]

fcomi st, st(1)

jb m\_sum

jae m\_cycle

m\_sum:

fadd st(2), st(0)

jmp m\_product

m\_product:

fmul st(3), st(0)

jmp m\_cycle

m\_cycle:

inc esi

fstp temp

fstp temp

loop m\_beg\_m

fst sum

fxch st(1)

fst product

fxch st(1)

add ebx, n\_size

pop ecx

loop m\_beg\_n

jmp m\_exit

m\_exit:

mov eax, 4C00h

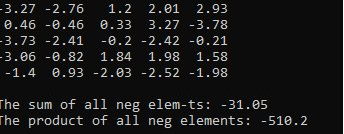
}

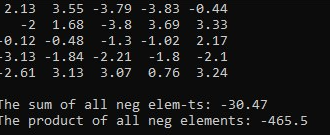
cout << endl << "The sum of all neg elem-ts: " << sum << endl << "The product of all neg elements: " << product << endl;

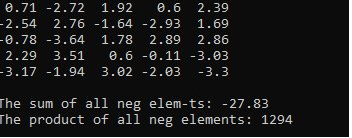
return 0;

}

**Результаты выполнения программы**

1) 

2) 

3) 

**Вывод:**

Я ознакомился с представлением одно- и многомерных массивов в памяти. Задал матрицу целочисленных и вещественных данных, а также реализовал ее обработку согласно своему варианту на языке Ассемблера.