**Лабораторная работа №7**

**Тема лабораторной работы**

Функции. Введение в структуры.

**Задача 1**

**Постановка задачи**

Вычислить минимальное значение (одно) из положительных элементов каждого из двух массивов чисел M[11] и N[11], в которых могут присутствовать отрицательные числа

**Математическая модель**

**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| f | функция | int |
| k | размер массива | int |
| m1 | промежуточная переменная | int |
| m2 | промежуточная переменная | int |
| \*m | указатель | int |
| \*n | указатель | int |

**Код программы**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int f(int\* a, int n)

{

int min = abs(a[0]);

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

if(a[i] >= 0 && a[i] < min) min = a[i];

return min;

}

void array(int\* a, int n)

{

for(int i = 0; i < n; i++) scanf("%d", &a[i]);

}

int main(void)

{

int k = 11, m1, m2;

int \*m = (int\*) malloc(k\*sizeof(int));

int \*n = (int\*) malloc(k\*sizeof(int));

printf("M[11]\n");

array(m, k);

printf("N[11]\n");

array(n, k);

m1 = f(m, k);

m2 = f(n, k);

printf("\n");

printf("%d\n%d", m1, m2);

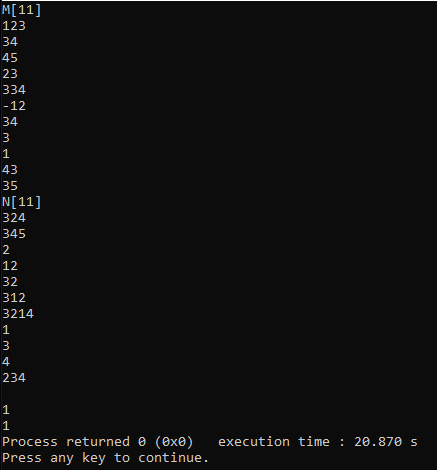
free(m);

free(n);

return 0;

}

**Результат выполненной работы**

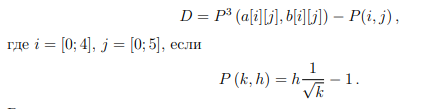
****

**Задача 2**

**Постановка задачи**

Вычислить

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| n | кол-во строк матрицы | int |
| m | кол-во столбцов матрицы | int |
| Free | функция | void |
| \*\*memory | указатель | int |
| \*\*A | указатель | int |
| i | параметр цикла | int |
| j | параметр цикла | int |

**Код программы**

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#define n 3

#define m 2

void Free(int \*\*A) {

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

free(A[i]);

free(A);

}

int\*\* memory(char abc[]) {

int\*\* A = (int\*\*)malloc(n \* sizeof(int\*));

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

A[i] = (int\*)malloc(m \* sizeof(int));

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

for (int j = 0; j < m; j++) {

printf("%s[%d][%d] = ", name, i, j);

scanf("%d", &A[i][j]);

}

return A;

}

double P(int k, int h) {

return (h \* 1 / sqrt(k) - 1);

}

int main(void) {

int \*\*a = memory("a");

int \*\*b = memory("b");

double D;

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

for (int j = 0; j < m; j++) {

if (i != 0 && a[i][j] != 0) {

D = pow(P(a[i][j], b[i][j]), 3) - P(i, j);

printf("For i = %d, j = %d, D = %lf\n", i, j, D);

}

else printf("Error. Divide by zero\n");

}

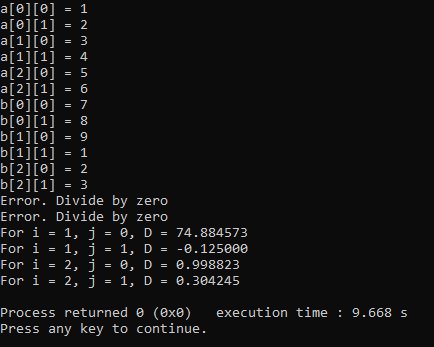
Free(a);

Free(b);

return 0;

}

**Результат выполненной работы**

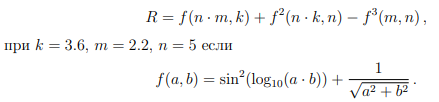


**Задача 3**

**Постановка задачи**

Вычислить

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| f(a,b) | функция | double |
| k | константа |
| m | константа |
| n | константа |
| f1 | промежуточная переменная |
| f2 | промежуточная переменная |
| f3 | промежуточная переменная |

**Код программы**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

double f(double a, double b)

{

return(pow(sin(log10(a\*b)), 2)+1/sqrt(a\*a+b\*b));

}

int main(void)

{

double k = 3.6, m = 2.2, n = 5, f1, f2, f3;

f1 = f(n\*m, k);

f2 = f(n\*k, n);

f3 = f(m, n);

printf("%f", f1+f2\*f2+pow(f3, 3));

return 0;

}

**Результат выполненной работы**



**Задача 5**

**Постановка задачи**

Даны две квадратные матрицы A и B размером N × N каждая. Напечатать ту из них, которая имеет минимальный «след» (т.е. сумму элементов главной диагонали). Математически это обозначается Tr(A) (англ. - trace) или Sp(A) (нем. - spur). При решении создать функцию для нахождения следа матрицы и функцию печати матрицы.

**Математическая модель**



**Список идентификаторов**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| f(m,n) | функция | int |
| Free(\*\*a,n) | функция | void |
| matrix(\*\*a,m,n) | функция | void |
| s(\*\*a,m) | функция | int |
| \*\*a | указатель | int |
| \*\*b | указатель |
| as | промежуточная переменная |
| bs | промежуточная переменная |
| n | вводимое с клавиатуры число |

**Код программы**

#include <stdio.h>

#include <stdlib.h>

int\*\* f(int m, int n)

{

int \*\*a;

a = (int\*\*) malloc(m\*sizeof(int));

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

a[i] = (int\*) malloc(n\*sizeof(int));

return a;

}

void Free(int \*\*a, int n)

{

int i;

for(i = 0; i < n; i++)

free(a[i]);

free(a);

}

void matrix(int \*\*a, int m, int n)

{

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

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

{

scanf("%d", &a[i][j]);

}

}

int s(int \*\*a, int m)

{

int sum = 0;

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

sum += a[i][i];

return sum;

}

int main(void)

{

int \*\*a, \*\*b, n, as, bs;

printf("Matrix size: ");

scanf("%d", &n);

a = f(n, n);

b = f(n, n);

printf("1st matrix:\n");

matrix(a, n, n);

printf("2nd matrix:\n");

matrix(b, n, n);

as = s(a, n);

bs = s(b, n);

if(as < bs)

printf("Minimum trace in the 1st matrix : %d", as);

else

printf("Minimum trace in the 2nd matrix : %d", bs);

Free(a, n); Free(b, n);

return 0;

}

**Результат выполненной работы**

