**Лабораторная работа 6  
Динамические массивы**

**Цель работы**: изучить работу динамических массивов в языке программирования С.

**Задание 1**

**Постановка задачи:**

Создать динамический одномерный массив целых чисел F размерности 12 и заполнить его положительными и отрицательными числами. , где R — сумма отрицательных элементов F, Q — отрицательный элемент массива, S — произведение положительных элементов массива F.

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#include <malloc.h>

int N = 12;

int main() {

srand(time(NULL));

int\* F = (int\*)malloc(N \* sizeof(int\*));

int R = 0;

int Q = 0;

int S = 1;

printf("Array F:\n");

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

F[i] = rand() % 100 - 10;

printf("%i ", F[i]);

if (F[i] < 0) {

R += F[i];

if(Q = 0)

Q = F[i];

}

else

S \*= F[i];

}

free(F);

float T = (float)(R + Q + S) / (float)(R \* Q \* S + 2);

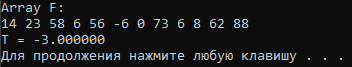
printf("\nT = %f\n", T);

system("pause");

return 0;

}

**Вывод программы:**



**Задание 2**

**Постановка задачи:**

Дан одномерный массив целых чисел A размера 12. Вычислить: Y = (U + T) · (S + 2), где S — количество элементов массива A с нечётными индексами, T — наибольший по модулю элемент массива A, U — сумма отрицательных элементов массива A.

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#include <malloc.h>

int N = 12;

int main() {

srand(time(NULL));

int S = 0;

int T = 0;

int U = 0;

int Y;

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

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

A[i] = rand() % 50 - 50;

printf("%d ", A[i]);

if (i % 2 == 1)

S += 1;

if (abs(A[i]) > T)

T = abs(A[i]);

if (A[i] < 0) U += A[i];

}

free(A);

printf("\nS = %d\nT = %d\nU = %d\n", S, T, U);

Y = (U + T) \* (S + 2);

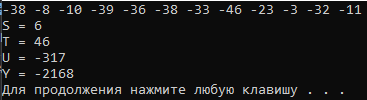
printf("Y = %d\n", Y);

system("pause");

return 0;

}

**Вывод программы:**



**Задание 3**

**Постановка задачи:**

Напишите программу для вычисления пересечения двух конечных множеств (наборов) A и B целых чисел одинакового размера с использованием динамических массивов. В качестве множества A можно взять первые 12 чисел ряда Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144. В качестве множества B можно взять первые 12 чисел последовательности Падована: 1, 1, 1, 2, 2, 3, 4, 5, 7, 9, 12, 16. Дублирующиеся значения можно исключать.

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#include <malloc.h>

int main() {

int N;

printf("Enter N: ");

scanf("%d", &N);

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

int \*B = (int\*)malloc(N \* sizeof(int));

A[0] = 1;

A[1] = 1;

for (int i = 2; i < N; i++){

A[i] = A[i - 2] + A[i - 1];

}

B[0] = 1;

B[1] = 1;

B[2] = 1;

for (int i = 3; i < N; i++){

B[i] = B[i - 2] + B[i - 3];

}

printf("\nA: ");

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

printf("%i ", A[i]);

printf("\n");

printf("B: ");

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

printf("%i ", B[i]);

printf("\n");

printf("Intersection AnB: ");

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

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

{

if (B[i] == A[j])

printf("%i ", B[i]);

}

printf("\n");

free(A);

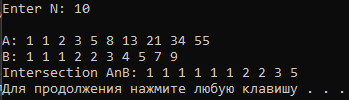
free(B);

system("pause");

return 0;

}

**Вывод программы:**



**Задание 4**

**Постановка задачи:**

Выделить динамически память под некоторую матрицу A размерности M × N и заполнить её произвольными числами. Сократить правильно размер этой матрицы, удалив из неё одну выбранную строку, освободив от неё также и память, используя указатели. После удаления строки в матрице AM×N должна быть возможность обхода всех элементов «новой» матрицы AM−1×N таким же способом, что и изначальной матрицы. Распечатать (используя циклы) матрицу до удаления строки и после удаления.

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#include <malloc.h>

int main() {

int m, n, u;

printf("Enter the number of rows(m) and columns(n): ");

scanf("%i%i", &m, &n);

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

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

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

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

A[i][j] = rand() % 5;

printf("%i ", A[i][j]);

}

printf("\n");

}

printf("Enter the number of string that need delete: ");

scanf("%i", &u);

free(A[u - 1]);

for (int i = u - 1; i < m - 1; i++){

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

A[i][j] = A[i + 1][j];

}

for (int i = 0; i < m - 1; i++){

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

printf("%i ", A[i][j]);

printf("\n");

}

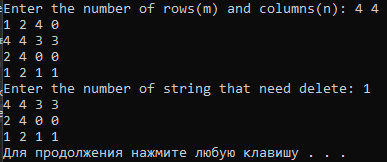
free(A);

system("pause");

return 0;

}

**Вывод программы:**



**Задание 5**

**Постановка задачи:**

Написать программу, которая вычисляет некоторый вектор b, как результат умножения некоторой матрицы M на вектор a: b = M × a .

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#include <malloc.h>

int main() {

srand(time(NULL));

int N;

printf("Enter the number of elements: ");

scanf("%i", &N);

int \*a = (int\*)calloc(N, sizeof(int));

int \*b = (int\*)calloc(N, sizeof(int));

int \*\*C = (int\*)malloc(N \* sizeof(int));

printf("Vector a:\n");

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

a[i] = rand() % 5;

printf("%i ", a[i]);

}

printf("\n");

printf("Matrix C:\n");

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

C[i] = (int\*)malloc(N \* sizeof(int));

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

C[i][j] = rand() % 5;

printf("%i ", C[i][j]);

}

printf("\n");

}

printf("Vector b:\n");

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

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

{

b[j] = C[i][j] \* a[i];

printf("%i ", b[j]);

}

}

printf("\n");

free(a);

free(b);

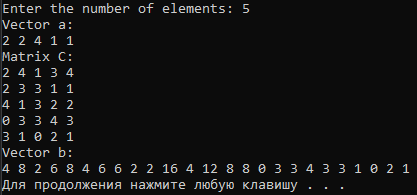
free(C);

system("pause");

return 0;

}

**Вывод программы:**



**Задание 6**

**Постановка задачи:**

Напишите программу, в которой создаётся квадратная матрица, заполненная нулями и единицами. Единичные значения у тех элементов, для которых сумма индексов является нечётным числом. Нулевые значения у тех элементов, для которых сумма индексов является чётным числом.

**Код программы:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include "time.h"

#include <malloc.h>

int main() {

int n = 0;

printf("Enter the size of square matrix: ");

scanf("%i", &n);

int \*\*A = (int\*)calloc(n, sizeof(int\*));

printf("Matrix A:\n");

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

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

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

if ( (i+j) % 2 == 0)

A[i][j] = 0;

else

A[i][j] = 1;

printf("%i ", A[i][j]);

}

printf("\n");

}

system("pause");

return 0;

}

**Вывод программы:**

