Лабораторная работа №6.

**Тема: Динамические массивы.**

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.

**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| **\*f** | Указатель на массив | **int** |
| **i** | Счетчик цикла | **int** |
| **s** | Произведение положительных чисел | **int** |
| **q** | Отрицательный (первый) элемент массива | **int** |
| **r** | Сумма отрицательных элементов массива | **int** |
| **t** | Результат | **float** |

**Код программы:**

#include <stdio.h>

#include <malloc.h>

int main()

{

int \*f, s = 1, r = 0, q = 0;

float t;

f = (int\*)calloc(12, sizeof(int));

for (int i = 0; i <12; i++)

{

f[i] = rand() % 10 - 5;

printf("%3d ",f[i]);

if (f[i]< 0)

{

if (q == 0)

{

q = f[i];

}

r += f[i];

}

if (f[i] > 0)

{

s \*= f[i];

}

}

t = (r + q + s)\*1.0 / (r\*q\*s + 2);

printf("\nt = %f", t);

free(f);

getchar();

return 0;

}

**Результат:**

****

Задание №2.

**Постановка задачи:**

Дан одномерный массив целых чисел A размера 12. Вычислить:

****

где S — количество элементов массива A с нечётными индексами, T — наибольший по модулю элемент массива A, U — сумма отрицательных элементов массива A.

**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| **\*f** | Указатель на массив | **int** |
| **i** | Счетчик цикла | **int** |
| **s** | Количество элементов массива с нечетными индексами | **int** |
| **t** | Наибольший по модулю элемент массива | **int** |
| **u** | Сумма отрицательных элементов массива | **int** |
| **y** | Результат | **int** |

**Код программы:**

#include <stdio.h>

#include <malloc.h>

#include <math.h>

int main()

{

int \*f, s=0, t=0, u=0, y;

f = (int\*)calloc(12,sizeof(int));

for (int i = 0; i < 12; i++)

{

f[i] = rand() % 20 - 10;

printf("%3d ", f[i]);

if (i % 2 == 1) s+=1;

if (abs(f[i]) > t) t = abs(f[i]);

if (f[i] < 0) u += f[i];

}

printf("\ns = %d", s);

printf("\nt = %d", t);

printf("\nu = %d", u);

y = (u + t)\*(s + 2);

printf("\ny = %d", y);

free(f);

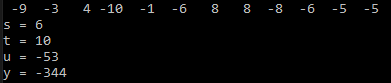
getchar();

getchar();

return 0;

}

**Результат:**



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.

**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| **\*f** | Указатель на массив чисел Фибоначчи | **int** |
| **\*p** | Указатель на массив чисел Падована | **int** |
| **i, j** | Счетчик цикла | **int** |
| **n** | Количество элементов массива | **int** |

**Код программы:**

#include <stdio.h>

#include <malloc.h>

#include <locale.h>

int main()

{

char \*locale = setlocale(LC\_ALL, "");

int \*f, \*p, n;

printf("Введите кол-во элементов последовательностей ");

scanf("%d", &n);

printf("\n");

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

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

f[0] = 1;

f[1] = 1;

for (int i = 2; i <n; i++)

{

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

}

p[0] = 1;

p[1] = 1;

p[2] = 1;

for (int i = 3; i < n; i++)

{

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

}

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

{

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

{

if (p[i] == f[j]) printf("p[%d] = f[%d] = %d = %d\n", i, j, p[i], f[j]);

}

}

free(p);

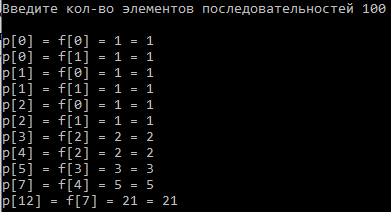
free(f);

getchar();

return 0;

}

**Результат:**



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.

**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| **\*\*a** | Указатель на матрицу | **int** |
| **i, j** | Счетчик цикла | **int** |
| **n, m** | Количество элементов массива | **int** |
| **c** | Строка для удаления | **int** |

**Код программы:**

#include <stdio.h>

#include <malloc.h>

#include <locale.h>

int main()

{

char \*locale = setlocale(LC\_ALL, "");

int \*\*a, m, n;

printf("Введите кол-во элементов строк и столбцов ");

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

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

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

{

printf("\n");

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

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

{

a[i][j] = rand() % 10;

printf("%3d", a[i][j]);

}

}

int c;

printf("\n\nВведите номер строки, которую необходимо удалить ");

scanf("%d", &c);

free(a[c-1]);

for (int i = c-1; i < n-1; i++)

{

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

{

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

}

}

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

{

printf("\n");

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

{

printf("%3d", a[i][j]);

}

}

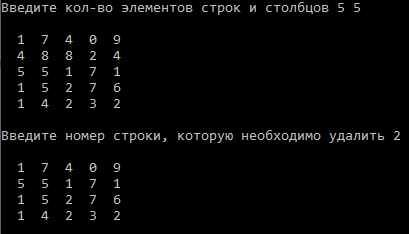
free(a);

getchar();getchar();

return 0;

}

**Результат:**



Задание №5.

**Постановка задачи:**

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

****

**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| **\*a** | Указатель на произвольный вектор | **int** |
| **\*\*m** | Указатель на массив чисел | **int** |
| **\*b** | Указатель на конечный вектор умножения | **int** |
| **i, j** | Счетчик цикла | **int** |
| **n** | Количество элементов массива | **int** |

**Код программы:**

#include <stdio.h>

#include <malloc.h>

#include <locale.h>

int main()

{

char \*locale = setlocale(LC\_ALL, "");

int \*\*m, \*b, \*a, n;

printf("Введите кол-во элементов ");

scanf("%d", &n);

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

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

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

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

{

printf("\n");

a[i] = rand() % 10;

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

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

{

m[i][j] = rand() % 10;

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

printf("%3d", b[j]);

}

}

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

free(m[i]);

free(m);

free(a);

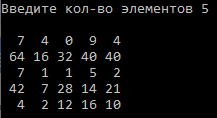
free(b);

getchar();getchar();

return 0;

}

**Результат:**



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.

**Список идентификаторов:**

| **Имя** | **Смысл** | **Тип** |
| --- | --- | --- |
| **\*\*p** | Указатель на массив | **int** |
| **i, j** | Счетчик цикла | **int** |
| **n** | Количество элементов массива | **int** |

**Код программы:**

#include <stdio.h>

#include <malloc.h>

#include <locale.h>

int main()

{

char \*locale = setlocale(LC\_ALL, "");

int \*\*p, n;

printf("Введите кол-во элементов ");

scanf("%d", &n);

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

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

{

printf("\n");

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

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

{

if ((i + j) % 2 == 0) { p[i][j] = 0; }

else { p[i][j] = 1; }

printf("%3d", p[i][j]);

}

}

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

free(p[i]);

free(p);

getchar();getchar();

return 0;

}

**Результат:**

