Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 15

По дисциплине «Основы алгоритмизации и программирования»

На тему «Динамическое выделение памяти»

Выполнил:

Студент 1 курса 10 группы

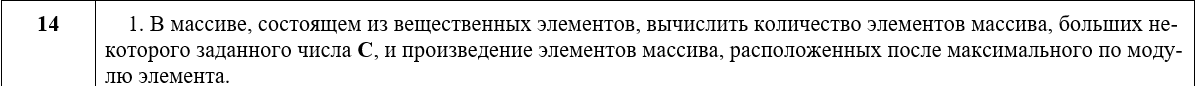
Сегренёв Кирилл Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

**ВАРИАНТ 14**

**Задание 1.**



**Код:**

#include <stdio.h>

#include <stdlib.h>

#include <ctime>

#include <iostream>

// Вариант 14

using namespace std;

void main()

{

setlocale(LC\_ALL, "ru");

srand(time(0));

float\* arr;

int n, idMax, C, col = 0;

float max = 0, perem = 1;

cout << "Введите размер массива: ";

cin >> n;

cout << "Введите число, которое будет разграничивать массив: ";

cin >> C;

if (!(arr = (float\*)malloc(n \* sizeof(int)))) // Объявляем наш массив через функцию malloc() и проверяем, есть ли место в памяти для него, или нет.

{

cout << "Not enough memory" << endl;

return;

}

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

{

arr[i] = static\_cast<float>(rand()) / RAND\_MAX \* 20; // В наш массив записываем рандомные числа от 0 до 20. Числа будут типа float

}

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

if (arr[i] > C) {

col++; // Считаем количество элементов, которые будут больше числа, введенного с клавиатуры

}

}

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

if (max < abs(arr[i])) {

max = abs(arr[i]);

idMax = i; // Находим индекс максимального элемента по модулю

}

}

// Выведем массив для удобства проверки задания

cout << "Массив: " << endl;

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

{

cout << arr[i] << ' ';

}

cout << '\n';

for (idMax = idMax + 1; idMax < n; idMax++) { // Вычисляем произведение чисел после максимального значения

perem \*= arr[idMax];

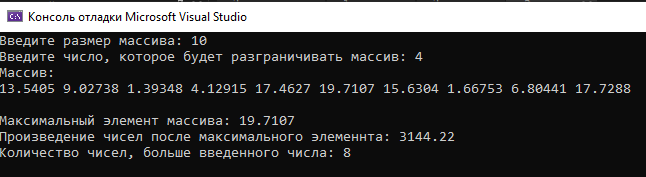
}

std::cout << '\n' << "Максимальный элемент масисва: " << max << '\n' << "Произведение чисел после максимального элеменнта: " << perem << '\n' << "Количество чисел, больше введенного числа: " << col << endl;

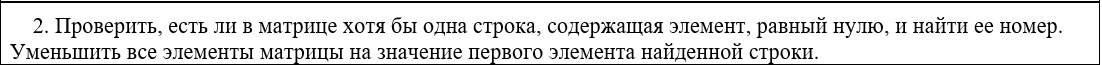
free(arr);

}

**Output:**



**Задание 2.**



**Код:**

#include <iostream>

#include <ctime>

// Вариант 14

using namespace std;

void main() {

setlocale(LC\_ALL, "ru");

srand(time(0));

int n, m, str = 0, zn;

cout << "Введите количество строк и столбцов матрицы(через пробел): ";

cin >> n >> m;

int\*\* arr = new int\* [n]; // Сначала объявляем массив для строк

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

arr[i] = new int[m]; // Для каждой строки объявляем новый массив, который будет заполнять значения данной строки

for (int i = 0; i < n; i++) // Записываем в матрицу рандомные значения

{

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

arr[i][j] = rand() % 21 - 10;

}

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

{

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

{

if (arr[i][j] == 0) { // Находим, в какой строке находится первый 0 и завершаем поиск

str = i + 1; // Добавляем к строке +1 для легкого восприятия данного значения пользователем

break;

}

}

if (str != 0) {

zn = arr[str - 1][0]; // Записываем в новую переменную значение первого элемента строки, где встречается 0

break;

}

}

if (str != 0) // Чтобы не возникала ошибка в коде добавляем условие, которое проверяет, был ли у нас 0 в матрице или нет

{

cout << "Матрица до изменеий: " << endl;

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

{

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

cout << arr[i][j] << "\t";

cout << "\n";

}

cout << '\n';

cout << "Матрица после изменеий: " << endl;

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

{

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

cout << arr[i][j] - zn << "\t";

cout << "\n";

}

cout << '\n' << "Номер строки, в которой встречается первый ноль: " << str << endl;

}

else

cout << "Нуля в матрице нет" << endl; // Если его не было, то выводим данное сообщение

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

{

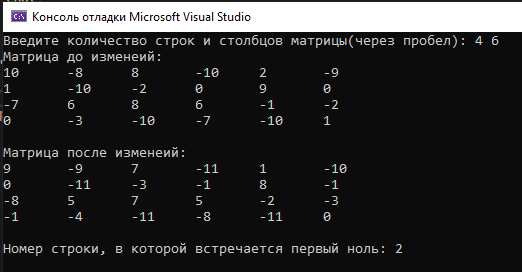
delete[] arr[i];

}

delete[] arr;

}

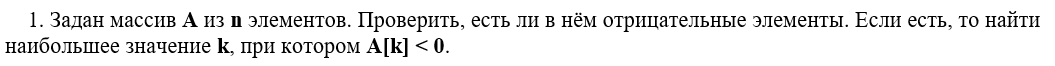
**Output:**



**ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ**

**ВАРИАНТ 2**

**Задание 1.**

****

**Код:**

#include <iostream>

#include <ctime>

// Вариант 2

using namespace std;

void main(){

setlocale(LC\_ALL, "ru");

srand(time(0));

int\* A;

int n, markIndex, mark = 0;

cout << "Введите дилну массива: ";

cin >> n;

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

if (!A) {

cout << "Not enough memory";

return;

}

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

{

A[i] = static\_cast<float>(rand()) / RAND\_MAX \* 20 - 10; // Записываем в массив рандомные числа

if (mark == 0) { // Для того, чтобы больше не входить в след условие, используем данное условие ( mark == 0 )

if (A[i] < 0) {

mark++; // Увеличиваем нашу переменную, которая отвечает за то, находится ли в массиве хотябы 1 отрицательный элемент

}

}

}

// В данном цикле находим индекс последнего отрицательного числа в массиве

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

if (A[i] < 0) {

markIndex = i;

}

}

cout << "Массив: " << '\n';

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

{

cout << A[i] << ' ';

}

cout << '\n';

if (mark != 0) { // Есть ли в массиве отрицательный элемент

cout << '\n' << "В данном массиве есть хотябы один отрицательный элемент" << '\n' << "Максимальный по индексу отрицательный элемент: " << markIndex + 1 << endl;

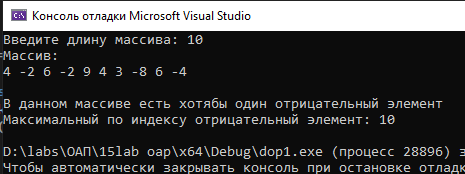
}

else

cout << "В матрице нет отрицательных элементов." << endl;

}

**Output:**

****

**Задание 2.**

****

**Код:**

#include <iostream>

#include <ctime>

// Вариант 2

using namespace std;

void main() {

setlocale(LC\_ALL, "ru");

srand(time(0));

int n, m, str = 0, max;

cout << "Введите количество строк и столбцов матрицы(через пробел): ";

cin >> n >> m;

int\*\* arr = new int\* [n]; // Сначала объявляем массив для строк

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

arr[i] = new int[m];

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

{

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

{

arr[i][j] = rand() % 101 - 10; // Записываем в матрицу рандомные значения

}

}

int strind, colind;

cout << "Матрица до изменений: " << endl;

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

{

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

cout << arr[i][j] << "\t";

cout << "\n";

}

cout << '\n';

max = arr[0][0]; // Присваиваем переменной max значение 1 элемента матрицы и с помощью ее находим максимальный элемент матрицы в след. цикле

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

{

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

if (max < arr[i][j]) {

max = arr[i][j];

strind = i; // В новые переменные записываем индексы элементы(строку и столбец)

colind = j;

}

}

}

swap(arr[0][0], arr[strind][colind]); // Меняем местами первый элемент и максимальный элемент матрицы

cout << "Матрица после изменений: " << endl;

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

{

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

cout << arr[i][j] << "\t";

cout << "\n";

}

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

{

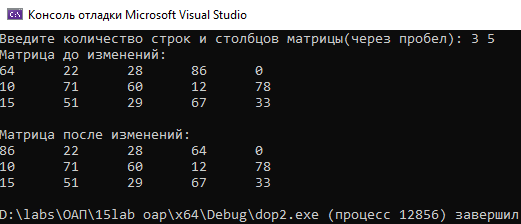
delete[] arr[i];

}

delete[] arr;

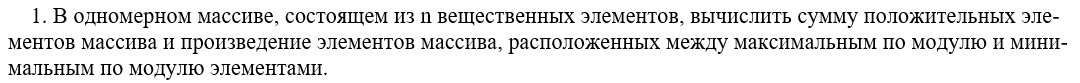
}

**Output:**



**ВАРИАНТ 10**

**Задание 1.**

****

**Код:**

#include <stdio.h>

#include <stdlib.h>

#include <ctime>

#include <iostream>

// Вариант 10

using namespace std;

void main()

{

setlocale(LC\_ALL, "ru");

srand(time(0));

int\* arr;

int n, idMax = 0, idMin = 0, sum = 0, multi = 1;

cout << "Введите размер массива: ";

cin >> n;

if (!(arr = (int\*)malloc(n \* sizeof(int)))) // Объявляем наш массив через функцию malloc() и проверяем, есть ли место в памяти для него, или нет.

{

cout << "Not enough memory" << endl;

return;

}

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

{

arr[i] = static\_cast<float>(rand()) / RAND\_MAX \* 41 - 20; // Записываем в массив рандомные числа

}

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

if (arr[i] > 0) {

sum += arr[i]; // Считаем сумму элементов, которые больше 0

}

}

int maxOrMin = arr[0];

for (int i = 0; i < n; i++) { // Находим индекс максимального элемента в массиве

if (maxOrMin < arr[i]) {

maxOrMin = arr[i];

idMax = i;

}

}

maxOrMin = arr[0];

for (int i = 0; i < n; i++) {// Находим индекс минимального элемента в массиве

if (maxOrMin > arr[i]) {

maxOrMin = arr[i];

idMin = i;

}

}

// Чтобы реализовать цикл, нужно понять, где стоят данные элементы относительно друг-друга, если idMin будет стоять левее idMax, то цикл будет одним

// Если правее, то другой

if (idMin - idMax < 0) {

for (++idMin; idMin < idMax; idMin++) {

multi \*= arr[idMin];

}

}

else

for (++idMax; idMax < idMin; idMax++) {

multi \*= arr[idMax];

}

// Выведем массив для удобства проверки задания

cout << "Массив: " << endl;

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

{

cout << arr[i] << ' ';

}

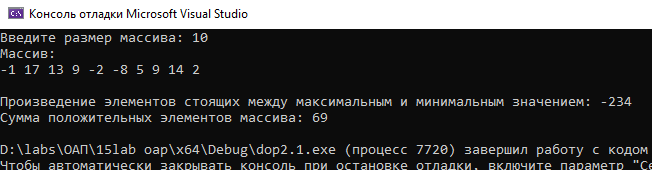
cout << '\n';

std::cout << '\n' << "Произведение элементов стоящих между максимальным и минимальным значением: " << multi << '\n' << "Сумма положительных элементов массива: " << sum << endl;

free(arr);

}

**Output:**

****

**Задание 2.**

****

**Код:**

#include <iostream>

#include <ctime>

// Вариант 10

using namespace std;

void main() {

setlocale(LC\_ALL, "ru");

srand(time(0));

int n, m, str = 0;

cout << "Введите количество строк и столбцов матрицы(через пробел): ";

cin >> n >> m;

int\*\* arr = new int\* [n]; // Сначала объявляем массив для строк

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

arr[i] = new int[m]; // Для каждой строки объявляем новый массив, который будет заполнять значения данной строки

for (int i = 0; i < n; i++) // Записываем в матрицу рандомные значения

{

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

arr[i][j] = rand() % 21 - 15;

}

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

{

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

{

if (arr[i][j] < 0) { // Проверяем, чтобы каждый элемент строки был меньше 0

str ++;

}

}

if (str == m) { // Если str равно количеству столбцов, то мы нашли строку, в которой все элементы отрицательны

str = i;

break; // Выходим из цикла, т.к как нужно найти первую строку отрицательных элементов

}

str = 0;

}

int zn = arr[str][0]; // Присваиваем переменной zn значение первого элемента найденой строки

if (str != 0) // Чтобы не возникала ошибка в коде добавляем условие, которое проверяет, был ли у нас вообще данная строка

{

cout << "Матрица до изменеий: " << endl;

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

{

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

cout << arr[i][j] << "\t";

cout << "\n";

}

cout << '\n';

cout << "Матрица после изменеий: " << endl;

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

{

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

cout << arr[i][j] + zn << "\t";

cout << "\n";

}

cout << '\n' << "Номер строки, в которой все элементы меньше 0: " << str + 1 << endl;

}

else

cout << "Строки с отрицательными элементами нет." << endl; // Если ее не было, то выводим данное сообщение

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

{

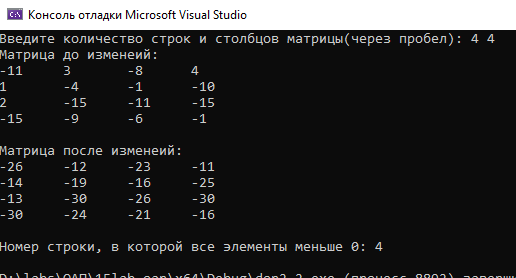
delete[] arr[i];

}

delete[] arr;

}

**Output:**



**ВАРИАНТ 12**

**Задание 1.**



**Код:**

#include <stdio.h>

#include <stdlib.h>

#include <ctime>

#include <iostream>

// Вариант 14

using namespace std;

void main()

{

setlocale(LC\_ALL, "ru");

srand(time(0));

int\* arr;

int n, col = 0;

cout << "Введите размер массива: ";

cin >> n;

if (!(arr = (int\*)malloc(n \* sizeof(int)))) // Объявляем наш массив через функцию malloc() и проверяем, есть ли место в памяти для него, или нет.

{

cout << "Not enough memory" << endl;

return;

}

for (int i = 0; i < n; i++) // Записываем в массив рандомные числа

{

arr[i] = rand() % 41 - 10;

}

int max = arr[0];

for (int i = 0; i < n; i++) { // Находим максимальный элемент и считаем, сколько раз он повторяется

if (arr[i] >= max) {

if (arr[i] = max) {

col++;

}

max = arr[i];

}

}

// Выведем массив для удобства проверки задания

cout << "Массив: " << endl;

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

{

cout << arr[i] << ' ';

}

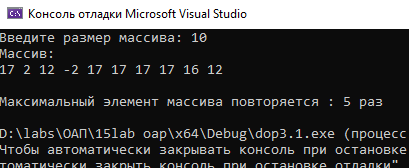
cout << '\n';

std::cout << '\n' << "Максимальный элемент массива повторяется : " << col << " раз" << endl;

free(arr);

}

**Output:**



**Задание 2.**

****

**Код:**

#include <iostream>

#include <ctime>

// Вариант 14

using namespace std;

void main() {

setlocale(LC\_ALL, "ru");

srand(time(0));

int n, m, str = 0;

cout << "Введите количество строк и столбцов матрицы(через пробел): ";

cin >> n >> m;

int\*\* arr = new int\* [n]; // Сначала объявляем массив для строк

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

arr[i] = new int[m]; // Для каждой строки объявляем новый массив, который будет заполнять значения данной строки

for (int i = 0; i < n; i++) // Записываем в матрицу рандомные значения

{

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

arr[i][j] = rand() % 21 - 18;

}

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

{

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

{

if (arr[i][j] > 0) { // Находим, в какой строке находится первый неотрицательный и завершаем поиск

str = i + 1; // Добавляем к строке +1 для легкого восприятия данного значения пользователем

break;

}

}

if (str != 0) {

break;

}

}

if (str != 0) // Чтобы не возникала ошибка в коде добавляем условие, которое проверяет, был ли у нас положительный элемент или нет

{

cout << "Матрица до изменеий: " << endl;

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

{

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

cout << arr[i][j] << "\t";

cout << "\n";

}

cout << '\n';

cout << "Матрица после изменеий: " << endl;

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

{

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

{

if (i == str - 2) // Изменение прошлой строчки матрицы

{

cout << arr[i][j] \* (-1) << "\t";

}

else

cout << arr[i][j] << "\t";

}

cout << "\n";

}

cout << '\n' << "Номер строки, в которой встречается первый неотрицательный элемент: " << str << endl;

}

else

cout << "Положительного элемента нет." << endl; // Если его не было, то выводим данное сообщение

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

{

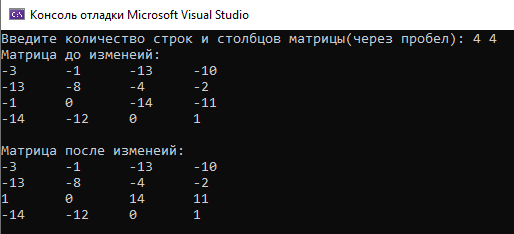
delete[] arr[i];

}

delete[] arr;

}

**Output:**

****