**Белорусский государственный технологический университет**

**Факультет информационных технологий**

**Кафедра программной инженерии**

Лабораторная работа 17

По дисциплине «Основы алгоритмизации и программирования»

На тему «Указатели и ссылки при работе с функциями»

Выполнил:

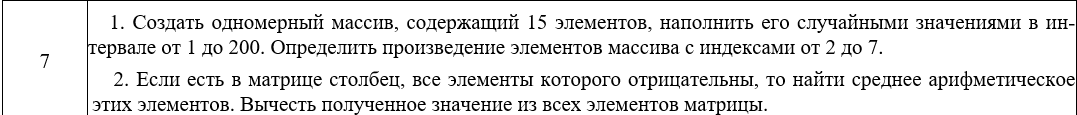
Студент 1 курса 10 группы

Мамонько Денис Александрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант 7**

****

**Задание 1**

**Код программы:**

//подключение библиотек

#include <iostream>

#include <ctime>

using namespace std;//чтобы не писать каждый раз std

//создание функции, которая будет считать произведение элементов, находящихся с 2 до 7

int massiv(int\* mas, int size) {

int result = 1;

for (int i = 2; i < 7; i++)

{

result \*= mas[i];//подсчет произведения

}

return result;//возвращение проиведения

}

int main()//объявляем главную функцию

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

int size;//переменная, отвечающая за размер массива

srand(time(0));//подключаем функцию для рандомайзера

cout << "Введите длину массива: ";//вводим длину массива

cin >> size;

int\* mas = new int[size];//создание динамического массива

//с помощью рандомайзера генерируем элементы массива

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

{

mas[i] = rand() % 200 + 1;

cout << mas[i] << ' ';

}

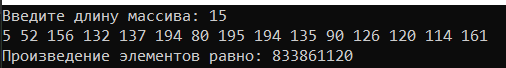
cout << endl;

cout << "Произведение элементов равно: " << massiv(mas, size);//возвращаемся к функции, считаем и выводим результат

delete[] mas;//очищаем массив из памяти

}

**Результат программы:**

****

**Задание 2**

**Код программы:**

#include <iostream>//покдлючение библиотеки

using namespace std;//чтобы не писать каждый раз std

//используем функцию для поиска отрицательных элементов

void Negative(int\*\* matrix, int n, int m) {

//с помощью флага находим отрицательный элемент

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

bool negative = true;

int sum = 0;

int count = 0;

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

if (matrix[i][j] >= 0) {

negative = false;

break;

}

sum += matrix[i][j];

count++;

}

if (negative) {

//если он отрицательный, то мы находим среднее арифметическое этих элементов и потом отнимаем это значение от всех других элементов матрицы

float sum1 = (float)sum / count;

cout << "Среднее арифметическое отрицательных элементов в столбце " << j + 1 << " равно: " << sum1 << endl;

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

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

matrix[i][k] -= sum1;

}

}

}

}

}

int main()//объявляем главную функцию

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

int n, m;//вводим размерность матрицы

cout << "Введите количество строк и столбцов: ";

cin >> n >> m;

int\*\* matrix = new int\* [n];//создание динамического массива

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

matrix[i] = new int[m];

cout << "Введите элементы матрицы: " << endl;//вводим элементы матрицы

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

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

cin >> matrix[i][j];

}

}

//возвращаемся к функции и выводим на экран изменненную матрицу

Negative(matrix, n, m);

cout << "Измененная матрица: " << endl;

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

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

for (int i = 0; i < n; i++)//очищаем массив из памяти

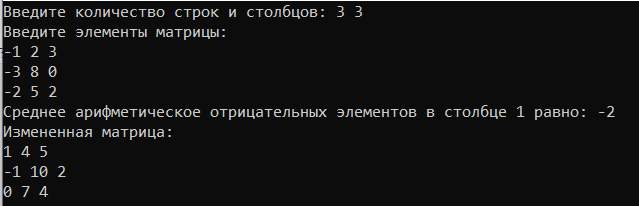
delete[] matrix[i];

delete[] matrix;

return 0;//завершаем функцию

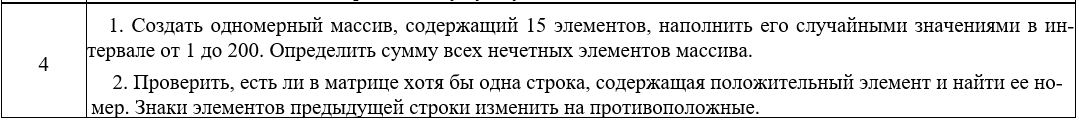
}

**Результат программы:**

****

**Дополнительные задания**

**Вариант 4**

****

**Задание 1**

**Код программы:**

//подключение библиотек

#include <iostream>

#include <ctime>

using namespace std;//чтобы не писать каждый раз std

//используем функцию для нахождения суммы нечетных элементов

int massiv(int\* mas, int size) {

int sum = 0;

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

{

if (mas[i] % 2 != 0) {

sum += mas[i];//находим сумму

}

}

return sum;//возвращаем сумму

}

int main()//объявляем главную функцию

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

int size;//инициализируем переменную для длины массива

srand(time(0));//используем функцию для рандомайзера

cout << "Введите длину массива: ";

cin >> size;//вводим длину массива

int\* mas = new int[size];//создание динамического массива

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

{

mas[i] = rand() % 200 + 1;//с помощью рандомайзера заполняем массив

cout << mas[i] << ' ';

}

cout << endl;

cout << "Сумма нечетных элементов равна: " << massiv(mas, size);//выводим сумму

delete[] mas;//очищаем массив из памяти

}

**Результат программы:**

****

**Задание 2**

**Код программы:**

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

//создание функции для поиска положительного элемента и его номера

void Positive(int\*\* matrix, int n, int m) {

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

bool positive = false;//осуществляем поиск с помощью флагов

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

if (matrix[i][j] > 0) {

positive = true;

break;

}

}

if (positive) {

cout << "Номер строки: " << i + 1 << endl;//выводим номер строки

if (i > 0) {

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

matrix[i - 1][k] = -matrix[i - 1][k];//меняем знаки на противоположные

}

}

break;

}

}

}

int main()//объявляем главную функцию

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

int n, m;//вводим размерность матрицы

cout << "Введите количество строк и столбцов: ";

cin >> n >> m;

int\*\* matrix = new int\* [n];//создание динамического массива

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

matrix[i] = new int[m];

cout << "Введите элементы матрицы: " << endl;//вводим элементы массива

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

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

cin >> matrix[i][j];

}

}

//выводим измененную матрицу на экран

Positive(matrix, n, m);

cout << "Измененная матрица: " << endl;

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

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

//очищаем массив из памяти

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

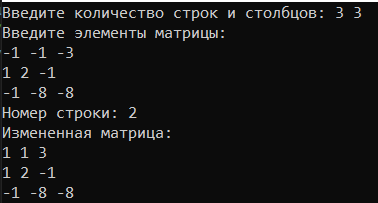
delete[] matrix[i];

delete[] matrix;

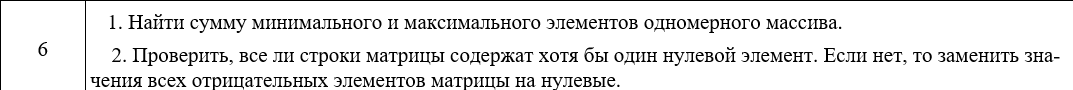
return 0;//завершение программы

}

**Результат программы:**

****

**Вариант 6**

****

**Задание 1**

**Код программы:**

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

//создание функции для нахождения сумму максимального и минимального элементов одномерного массива

int massiv(int\* mas, int size) {

int min = mas[0], max = mas[0], sum = 0;

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

{

if (mas[i] < min) {

min = mas[i];//находим минимальный элемент

}

if (mas[i] > max) {

max = mas[i];//находим максимальный элемент

}

if (max > mas[i] || min < mas[i]) {

sum = max + min;//находим сумму

}

}

return sum;//возвращаем сумму

}

int main()//объявляем главнуюб функцию

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

int size;//вводим длину массива

cout << "Введите длину массива: ";

cin >> size;

int\* mas = new int[size];//создание динамического массива

cout << "Введите элементы массива: ";

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

{

cin >> mas[i];//вводим элементы массива

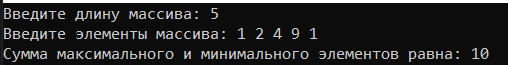
}

cout << "Сумма максимального и минимального элементов равна: " << massiv(mas, size);//выводим сумму

delete[] mas;//очищаем массив из памяти

}

**Результат программы:**

****

**Задание 2**

**Код программы:**

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

//создание функции для поиска нулевых элементов

void Zero(int\*\* matrix, int n, int m) {

bool zero = true;//осуществляем поиск с помощью флагов

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

bool zero1 = false;

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

if (matrix[i][j] == 0) {

zero1 = true;

break;

}

}

if (!zero1) {

zero = false;

break;

}

}

if (!zero) {//если элемент не равен 0, то меняем все отрицательные значения на нулевые

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

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

if (matrix[i][j] < 0) {

matrix[i][j] = 0;

}

}

}

}

}

int main()//объявляем главную функцию

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

int n, m;//вводим размерность матрицы

cout << "Введите количество строк и столбцов: ";

cin >> n >> m;

int\*\* matrix = new int\* [n];//создание динамического массива

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

matrix[i] = new int[m];

cout << "Введите элементы матрицы: " << endl;//вводим элементы матрицы

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

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

cin >> matrix[i][j];

}

}

//выводим измененную матрицу на экран

Zero(matrix, n, m);

cout << "Измененная матрица: " << endl;

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

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

//очищаем массив из памяти

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

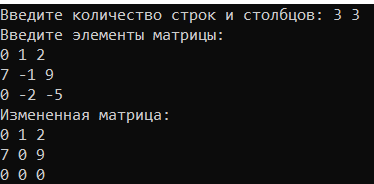
delete[] matrix[i];

delete[] matrix;

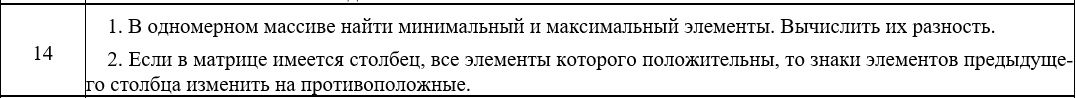
return 0;//завершение программы

}

**Результат программы:**

****

**Вариант 14**

****

**Задание 1**

**Код программы:**

#include <iostream>//подключаем библиотеку

using namespace std;//чтобы не писать каждый раз std

//создаем функцию, которая будет находить разность между максимальным и минимальным элементами

int massiv(int\* mas, int size) {

int min = mas[0], max = mas[0], diff = 0;

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

{

if (mas[i] < min) {

min = mas[i];//находим минимальный элемент

}

if (mas[i] > max) {

max = mas[i];//находим максимальный элемент

}

if (max > mas[i] || min < mas[i]) {

diff = max - min;//находим разность элементов

}

}

return diff;//вызвращаем разность

}

int main()//объявляем главную функцию

{

setlocale(LC\_ALL, "RU");//подключаем кириллицу

int size;//вводим длину массива

cout << "Введите длину массива: ";

cin >> size;

int\* mas = new int[size];//создание динамического массива

cout << "Введите элементы массива: ";

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

{

cin >> mas[i];//вводим элементы массива

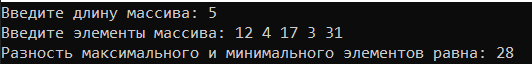
}

cout << "Разность максимального и минимального элементов равна: " << massiv(mas, size);//выводим разность

delete[] mas;//очищаем массив из памяти

}

**Результат программы:**

****

**Задание 2**

**Код программы:**

#include <iostream>//подключение библиотеки

using namespace std;//чтобы не писать каждый раз std

//создаем функцию для поиска положительных элементов

void Positive(int\*\* matrix, int n, int m) {

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

bool positive = false;//поиск осуществляем с помощью флагов

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

if (matrix[i][j] > 0) {

positive = true;

break;

}

}

if (positive && j > 0) {//если все элементы столбца положительные, то знаки элементов противоположного столбца меняем на противоположные

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

matrix[i][j - 1] = -matrix[i][j - 1];

}

}

}

}

int main()//объявляем главную функцию

{

setlocale(LC\_ALL, "RU");//подключение кириллицы

int n, m;//вводим размерность массива

cout << "Введите количество строк и столбцов: ";

cin >> n >> m;

int\*\* matrix = new int\* [n];//создание динамического массива

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

matrix[i] = new int[m];

cout << "Введите элементы матрицы: " << endl;//вводим элементы матрицы

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

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

cin >> matrix[i][j];

}

}

//выводим измененную функцию на экран

Positive(matrix, n, m);

cout << "Измененная матрица: " << endl;

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

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

cout << matrix[i][j] << " ";

}

cout << endl;

}

//очищаем массив из памяти

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

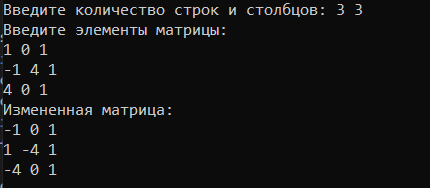
delete[] matrix[i];

delete[] matrix;

return 0;//завершение программы

}

**Результат программы:**

****