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»

Выполнила:

Студентка 1 курса 10 группы

Рублевская Маргарита Владимировна

Преподаватель: асс. Андронова М.В.

2023, Минск

**Вариант№ 13**

1. В соответствии со своим вариантом написать главную функцию, в которой имеются вызовы ***функций пользователя***, реализующих задачи из таблицы ниже. Ввод исходных данных и вывод результатов осуществить в главной функции, при этом использовать***динамические* *массивы***. Для передачи параметров в функции пользователя и возвращения результатов применить ***указатели*** и ***ссылки***.

1. Дан массив **A** вещественного типа, содержащий 20 положительных и отрицательных элементов. Сформировать массив **B** из положительных элементов массива **A**, имеющих четный индекс. Найти сумму квадратов элементов нового массива.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Функция для формирования массива B из положительных элементов массива A с четными индексами

void generateArrayB(const float\* A, int size, float\*& B, int& newSize) {

newSize = 0;

for (int i = 0; i < size; i += 2) {

if (A[i] > 0) {

newSize++;

}

}

B = new float[newSize];

int indexB = 0;

for (int i = 0; i < size; i += 2) {

if (A[i] > 0) {

B[indexB] = A[i];

indexB++;

}

}

}

// Функция для нахождения суммы квадратов элементов массива B

float sumOfSquares(const float\* B, int size) {

float sum = 0;

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

sum += (B[i] \* B[i]);

}

return sum;

}

int main() {

setlocale(LC\_CTYPE, "Russian");

const int SIZE = 20;

float\* A = new float[SIZE];

// Вводим исходные данные

std::cout << "Введите 20 положительных и отрицательных элементов массива A:\n";

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

std::cin >> A[i];

}

float\* B;

int newSize;

// Формируем массив B из положительные элементы массива A с четными индексами

generateArrayB(A, SIZE, B, newSize);

// Находим суммы квадратов элементов массива B

float result = sumOfSquares(B, newSize);

std::cout << "Массив B: ";

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

std::cout << B[i] << " ";

}

std::cout << std::endl;

// Выводим результат

std::cout << "Сумма квадратов элементов массива B: " << result << std::endl;

// Освобождение памяти, выделенной для массивов

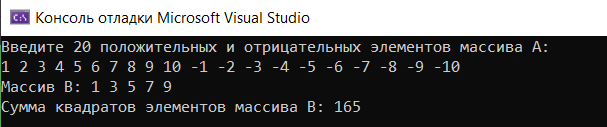
delete[] A;

delete[] B;

return 0;

}

Консоль VS:

****

2. Дана целочисленная квадратная матрица. Если минимальное значение находится на главной диагонали, то вывести его значение и индексы.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Функция для нахождения минимальное значение на главной диагонали квадратной матрицы и возвращению его значения и индекса строки, на которой оно находится.

void findMinValueAndIndex(int\*\* matrix, int size) {

int min = matrix[0][0];

int minIndex = 0;

for (int i = 1; i < size; i++) {

if (matrix[i][i] < min) {

min = matrix[i][i];

minIndex = i;

}

}

std::cout << "Минимальное значение на главной диагонали: " << min << std::endl;

std::cout << "Индексы: (" << minIndex << "," << minIndex << ")" << std::endl;

}

int main() {

setlocale(LC\_CTYPE, "Russian");

int size;

std::cout << "Введите размер квадратной матрицы: ";

std::cin >> size;

// Выделение памяти под двумерный динамический массив

int\*\* matrix = new int\* [size];

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

matrix[i] = new int[size];

}

// Заполняем массив значениями

std::cout << "Введите элементы матрицы:\n";

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

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

std::cin >> matrix[i][j];

}

}

// Вызываем функция для нахождения минимальное значение на главной диагонали квадратной матрицы

findMinValueAndIndex(matrix, size);

// Освобождение памяти, выделенной для массивов

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

delete[] matrix[i];

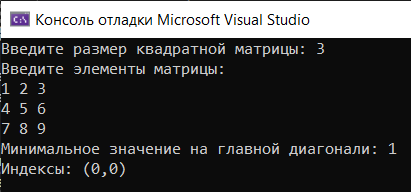
}

delete[] matrix;

return 0;

}

Консоль VS:

****

**Вариант№ 11**

1. Изменить одномерный массив, вычеркнув из него нечетные элементы.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Функция для удаления нечётных элементов из массива

void removeOddElements(int\*& array, int& size) {

int newSize = 0;

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

if (array[i] % 2 == 0) {

array[newSize] = array[i];

newSize++;

}

}

size = newSize;

}

int main() {

setlocale(LC\_CTYPE, "RU");

int size;

std::cout << "Введите размер массива: ";

std::cin >> size;

// Выделяем память под массив

int\* array = new int[size];

// Заполняем массив значениями

std::cout << "Введите элементы массива: " ;

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

std::cin >> array[i];

}

// Вызываем функцию для удаления нечётных элементов

removeOddElements(array, size);

// Выводим измененную матрицу на экран

std::cout << "Изменённый массив (без нечётных элементов): ";

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

std::cout << array[i] << " ";

}

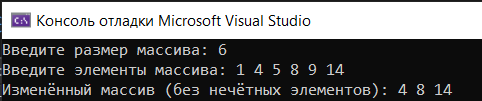
// Освобождаем память, выделенную для массива

delete[] array;

return 0;

}

Консоль VS:



2. Дана целочисленная квадратная матрица. Если она не содержат отрицательных элементов, то определить сумму элементов в тех строках, где отрицательные элементы отсутствуют.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Функция для определения суммы элементов в тех строках, где отрицательные элементы отсутствуют

void sumOfRowsWithoutNegatives(int\*\* matrix, int size) {

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

bool hasNegative = false;

int rowSum = 0;

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

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

hasNegative = true;

break;

}

rowSum += matrix[i][j];

}

// Если в строке нет отрицательных элементов, то выводии сообщение о сумме элементов в этой строке.

if (!hasNegative) {

std::cout << "Сумма элементов в строке " << i + 1 << ": " << rowSum << std::endl;

}

}

}

int main() {

setlocale(LC\_CTYPE, "RU");

int size;

std::cout << "Введите размер квадратной матрицы: ";

std::cin >> size;

// Выделение памяти под двумерный динамический массив

int\*\* matrix = new int\* [size];

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

matrix[i] = new int[size];

}

// Заполняем массив значениями

std::cout << "Введите элементы квадратной матрицы (" << size << "x" << size << "):" << std::endl;

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

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

std::cin >> matrix[i][j];

}

}

// Вызоваем функции для определения суммы элементов в тех строках, где отрицательные элементы отсутствуют

sumOfRowsWithoutNegatives(matrix, size);

// Освобождение памяти, выделенной для массива

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

delete[] matrix[i];

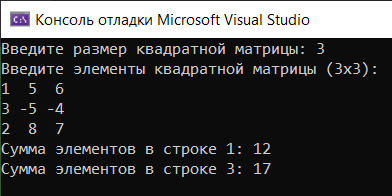
}

delete[] matrix;

return 0;

}

Консоль VS:



**Вариант№ 14**

1. В одномерном массиве найти минимальный и максимальный элементы. Вычислить их разность.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Функция для нахождения разницы между наибольшим и наименьшим элементами массива.

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

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

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

{

// Находим минимальный элемент

if (mas[i] < min) {

min = mas[i];

}

// Находим максимальный элемент

if (mas[i] > max) {

max = mas[i];

}

// Находим разность

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

diff = max - min;

}

}

return diff;

}

int main()

{

setlocale(LC\_CTYPE, "RU");

int size;

std::cout << "Введите размер массива: ";

std::cin >> size;

// Выделение памяти под массив целых чисел размера size.

int\* mas = new int[size];

// Заполняем массив значениями

std::cout << "Введите элементы массива: ";

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

{

std::cin >> mas[i];

}

// Находим разность

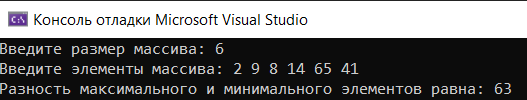
std::cout << "Разность максимального и минимального элементов равна: " << massiv(mas, size);

// Освобождение памяти, выделенной для массива

delete[] mas;

}

Консоль VS:



2. Если в матрице имеется столбец, все элементы которого положительны, то знаки элементов предыдущего столбца изменить на противоположные.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Функция меняет знаки элементов в каждом столбце, предшествующем столбцу, содержащему хотя бы одно положительное число.

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

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

bool positive = false;

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

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

positive = true;

break;

}

}

// Если все элементы столбца положительные, то знаки элементов противоположного столбца меняем на противоположные

if (positive && j > 0) {

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

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

}

}

}

}

int main()

{

setlocale(LC\_CTYPE, "RU");

int n, m;

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

std::cin >> n >> m;

// Выделение памяти под двумерный динамический массив

int\*\* matrix = new int\* [n];

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

matrix[i] = new int[m];

// Заполняем массив значениями

std::cout << "Введите элементы матрицы: " << std::endl;

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

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

std::cin >> matrix[i][j];

}

}

// Вызываем функцию, меняющую знаки элементов в каждом столбце, предшествующем столбцу, содержащему хотя бы одно положительное число.

Positive(matrix, n, m);

// Выводим измененную матрицу на экран

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

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

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

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

}

std::cout << std::endl;

}

// Освобождение памяти, выделенной для массива

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

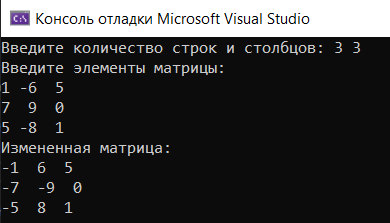
delete[] matrix[i];

delete[] matrix;

return 0;

}

Консоль VS:



**Вариант№ 6**

1. Найти сумму минимального и максимального элементов одномерного массива.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Функция для нахождения суммы между наибольшим и наименьшим элементами массива.

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

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

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

{

// Находим минимальный элемент

if (mas[i] < min) {

min = mas[i];

}

// Находим максимальный элемент

if (mas[i] > max) {

max = mas[i];

}

// Находим сумму

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

sum = max + min;

}

}

return sum;

}

int main()

{

setlocale(LC\_CTYPE, "RU");

int size;

std::cout << "Введите размер массива: ";

std::cin >> size;

// Выделение памяти под массив целых чисел размера size.

int\* mas = new int[size];

// Заполняем массив значениями

std::cout << "Введите элементы массива: ";

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

{

std::cin >> mas[i];

}

// Находим сумму

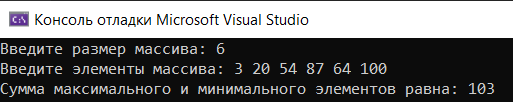
std::cout << "Сумма максимального и минимального элементов равна: " << massiv(mas, size);

// Освобождение памяти, выделенной для массива

delete[] mas;

}

Консоль VS:



2. Проверить, все ли строки матрицы содержат хотя бы один нулевой элемент. Если нет, то заменить значения всех отрицательных элементов матрицы на нулевые.

Код:

#include <iostream> // Подключаем библиотеку для работы с вводом/выводом

// Функция заменяет все отрицательные элементы на 0, если в матрице нет строки, содержащей только нули.

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

bool zero = true;

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()

{

setlocale(LC\_CTYPE, "RU");

int n, m;

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

std::cin >> n >> m;

// Выделение памяти под двумерный динамический массив

int\*\* matrix = new int\* [n];

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

matrix[i] = new int[m];

// Заполняем массив значениями

std::cout << "Введите элементы матрицы: " << std::endl;

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

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

std::cin >> matrix[i][j];

}}

// Вызываем функцию заменяющую все отрицательные элементы на 0, если в матрице нет строки, содержащей только нули

Zero(matrix, n, m);

// Выводим измененную матрицу на экран

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

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

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

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

}

std::cout << std::endl;

}

// Освобождение памяти, выделенной для массива

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

delete[] matrix[i];

delete[] matrix;

return 0;

}

Консоль VS:

