**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:

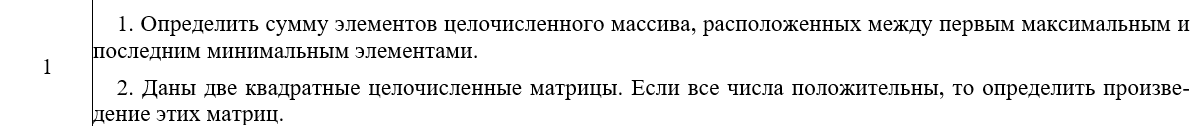
Студентка 1 курса 7 группы

Шинкевич Марина Дмитриевна

Преподаватель: асс. Андронова М.В.

2023, Минск

Дополнительные задания к лабораторной 17



#include <iostream>

#include <Windows.h>

using namespace std;

int& sum\_min\_max(int\* ptr, int& size, int& sum) {//Определяется функция sum\_min\_max, которая принимает указатель на массив целых чисел ptr, ссылку на переменную size (размер массива) и ссылку на переменную sum (сумма элементов массива).

int max = \*ptr, ind\_Max = 0, min = \*ptr, ind\_Min = 0;//Внутри функции определяются переменные max и min, которые инициализируются первым элементом массива, а также переменные ind\_Max и ind\_Min, которые будут хранить индексы максимального и минимального элементов.

for (int i = 0; i < size; i++) {//т цикл по массиву для поиска максимального и минимального элементов, а также их индексов.

if (\*(ptr + i) > max) {

max = \*(ptr + i);

ind\_Max = i;

}

if (\*(ptr + i) < min) {

min = \*(ptr + i);

ind\_Min = i;

}

}

//определяются переменные start и end, которые хранят индексы начала и конца участка массива между максимальным и минимальным элементами

int start = min(ind\_Max, ind\_Min);

int end = max(ind\_Max, ind\_Min);

for (int i = ++start; i < end; i++) {//цикл по этому участку массива для подсчета суммы его элементов, которая добавляется к переменной sum.

sum += \*(ptr + i);

}

return sum;

}

int main() {// В функции main создается массив ptr размером 15, заполняется случайными числами от 1 до 10, выводится на экран, затем вызывается функция sum\_min\_max для подсчета суммы элементов между минимальным и максимальным элементами, и эта сумма выводится на экран.

SetConsoleOutputCP(1251);

int size = 15, sum = 0;

int a = 1, b = 10;

int\* ptr = new int[size];

srand((unsigned)time(NULL));

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

ptr[i] = rand() % (b - a) + a;

cout << ptr[i] << ' ';

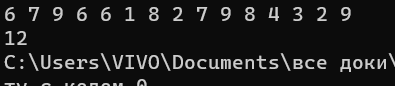
}

cout << '\n';

sum\_min\_max(ptr, size, sum);

cout << sum;

}



#include <iostream>

#include <ctime>

#include <Windows.h>

using namespace std;

int\*\* multiplication\_for\_matrix(int\*\* mat\_1, int\*\* mat\_2, int size) {

int\*\* res\_Matr = new int\* [size];//выделяет память под массив указателей на целые числа размером size.

for (int i = 0; i < size; i++) {//выделяет память для каждой строки матрицы размером size.

res\_Matr[i] = new int[size];

}

for (int i = 0; i < size; i++) {// вложенный цикл для умножения матриц по правилам матричного умножения.

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

res\_Matr[i][j] = 0;

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

res\_Matr[i][j] += mat\_1[i][k] \* mat\_2[k][j];

}

}

}

return res\_Matr;

}

int\*\* new\_matrix(int size, int a, int b) {//объявляет функцию, которая принимает размер матрицы size и диапазон случайных чисел [a, b]

int\*\* matrix = new int\* [size];// Выделяется память для матрицы matrix аналогично функции multiplication\_for\_matrix.

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

matrix[i] = new int[size];

}

srand((unsigned)time(NULL));//инициализирует генератор случайных чисел текущим временем.

for (int i = 0; i < size; i++) {//Заполняет матрицу случайными числами в заданном диапазоне и выводит ее на экран.

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

matrix[i][j] = rand() % (b - a) + a;

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

}

cout << '\n';

}

return matrix;

}

int main() {

SetConsoleOutputCP(1251);

int size = 4;

int a = 1, b = 15;//Определяет размер матрицы (size) и диапазон случайных чисел (a и b).

int\*\* mat\_1 = new\_matrix(size, a, b);//Создает две матрицы mat\_1 и mat\_2, используя функцию new\_matrix.

int\*\* mat\_2 = new\_matrix(size, a, b);

//Вызывает функцию multiplication\_for\_matrix для умножения матриц mat\_1 и mat\_2, и выводит результат на экран.

int\*\* res\_Matr = multiplication\_for\_matrix(mat\_1, mat\_2, size);

cout << "Результат :\n";

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

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

cout << res\_Matr[i][j] << ' ';

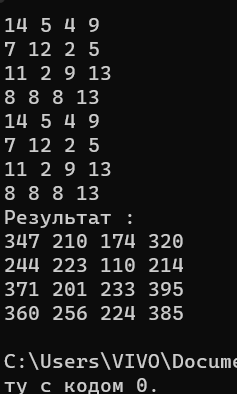
}

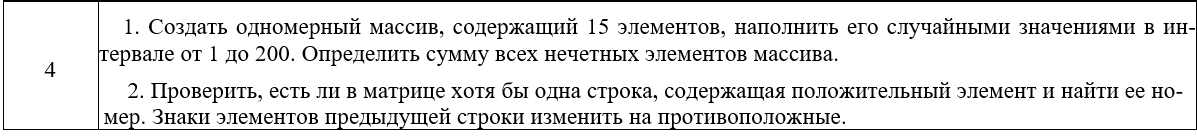
cout << '\n';

}

return 0;

}





#include <iostream>

#include <ctime>

#include <Windows.h>

using namespace std;

int& sum\_nechot(int\* array, int size, int& sum) { //Объявляет функцию sum\_nechot, которая принимает массив целых чисел (array), размер массива (size) и ссылку на целое число (sum), и возвращает ссылку на целое число.

sum = 0;

for (int i = 0; i < size; i++) {//Начинается цикл, который перебирает все элементы массива.

if (array[i] % 2 != 0) {// Проверяет, является ли текущий элемент массива нечётным (остаток от деления на 2 не равен 0). Если условие выполняется, то значение текущего элемента добавляется к переменной sum.

sum += array[i];

}

}

return sum;

}

int main() {

SetConsoleOutputCP(1251);

int size = 15, a = 1, b = 200;//Определяет размер массива (size) и диапазон случайных чисел (a и b).

int\* ptr = new int[size];// Выделяет память для массива целых чисел размером size.

srand((unsigned)time(NULL));//Инициализирует генератор случайных чисел текущим временем.

for (int i = 0; i < size; i++) {//В этом цикле заполняется массив случайными числами в заданном диапазоне и выводится на экран.

ptr[i] = rand() % (b - a) + a;

cout << ptr[i] << ' ';

}

cout << '\n';

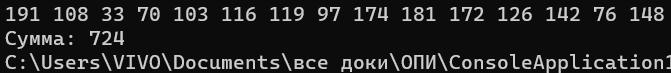
int sum = 0;

sum = sum\_nechot(ptr, size, sum);// Вызывает функцию sum\_nechot для подсчета суммы нечетных чисел в массиве ptr, передавая массив, его размер и переменную sum по ссылке.

cout << "Сумма: " << sum;

return 0;

}



#include <iostream>

#include <Windows.h>

using namespace std;

bool positive\_row(int\*\* matrica, int rows, int colums, int& row\_ind) {//Объявляет функцию positive\_row

bool bol = false;//Объявляет логическую переменную bol и устанавливает ее значение равным false.

for (int i = 0; i < rows; ++i) {//Начинается цикл, который перебирает строки матрицы.

for (int j = 0; j < colums; ++j) {//Вложенный цикл, который перебирает элементы внутри строки.

if (matrica[i][j] > 0) {//Проверяет, является ли текущий элемент матрицы положительным. Если условие выполняется, то устанавливает значение bol в true, записывает индекс строки с положительным числом в переменную row\_ind и прерывает цикл.

bol = true;

row\_ind = i;

break;

}

}

if (bol) {//Проверяет, было ли найдено положительное число в строке. Если да, то выполняет следующий блок кода.

if (row\_ind > 0) {// Проверяет, что индекс строки с положительным числом больше 0.

for (int j = 0; j < colums; ++j) {//Умножает все элементы строки перед строкой с положительным числом на -1

matrica[row\_ind - 1][j] \*= -1;

}

}

break;

}

}

return bol;

}

int main() {

SetConsoleOutputCP(1251);

int n = 10, m = 3, a = -10, b = 10;

int\*\* ptr = new int\* [n];//Выделяет память для двумерного массива целых чисел размером n x m.

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

ptr[i] = new int[m];

srand((unsigned)time(NULL));//Инициализирует генератор случайных чисел текущим временем.

for (int i = 0; i < n; i++) {//Заполняет матрицу случайными числами в заданном диапазоне и выводит ее на экран.

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

ptr[i][j] = rand() % (b - a) + a;

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

}

cout << '\n';

}

//Объявляет переменную row\_ind, которая будет содержать индекс строки с положительным числом, и логическую переменную Found\_Positive\_Num, которая будет хранить результат вызова функции positive\_row.

int row\_ind = 0;

bool Found\_Positive\_Num;

//Вызывает функцию positive\_row для поиска строки с положительным числом в матрице ptr, передавая матрицу, количество строк, количество столбцов и переменную row\_ind по ссылке.

Found\_Positive\_Num = positive\_row(ptr, n, m, row\_ind);

if (Found\_Positive\_Num) {//Выводит на экран информацию о найденной строке с положительным числом и новую матрицу после изменений, если такая строка была найдена.

cout << "Номер строки с положительным числом: " << row\_ind + 1 << endl;

cout << "\nНовая матрица: " << endl;

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

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

cout << ptr[i][j] << ' ';

}

cout << '\n';

}

}

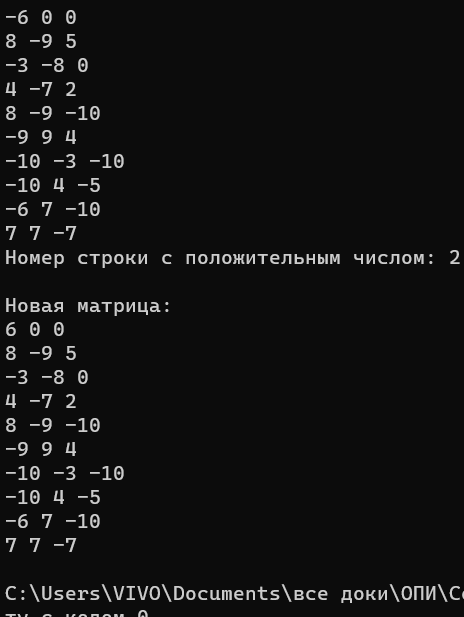
else {

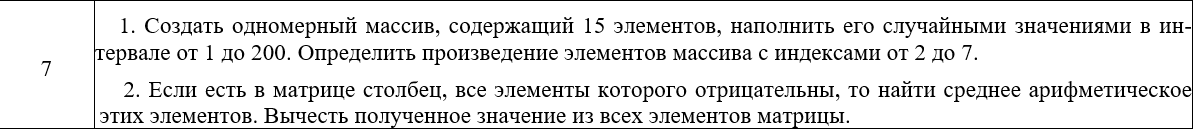
cout << "Числа не положительные";

}

return 0;

}





#include <iostream>

#include <Windows.h>

using namespace std;

long long& multiplication\_of\_numbers(int\* array, int size, long long& mult) {//Объявляет функцию multiplication\_of\_numbers

mult = 1;//Устанавливает значение переменной mult равным 1.

for (int i = 2; i <= 7; i++) {//цикл, который перебирает элементы массива с индексами от 2 до 7 включительно.

mult \*= array[i];// Умножает текущий элемент массива на значение переменной mult и сохраняет результат в mult.

}

return mult;

}

int main() {

SetConsoleOutputCP(1251);

int size = 15, a = 1, b = 200;

int\* ptr = new int[size];//Выделяет память для массива целых чисел размером size.

srand((unsigned)time(NULL));//Заполняет массив случайными числами в заданном диапазоне и выводит его на экран.

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

ptr[i] = rand() % (b - a) + a;

cout << ptr[i] << ' ';

}

cout << '\n';

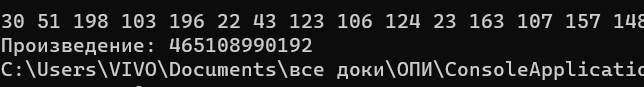
long long mult = 1;//Объявляет переменную mult, которая будет хранить произведение чисел.

mult = multiplication\_of\_numbers(ptr, size, mult);//Вызывает функцию multiplication\_of\_numbers для вычисления произведения элементов массива, передавая массив, его размер и переменную mult по ссылке.

cout << "Произведение: " << mult;

return 0;

}



#include <iostream>

#include <Windows.h>

using namespace std;

double& average\_values(double\*\* array, int x, int y, double& aver) {//Объявляет функцию average\_values

bool bol = false;//Объявляет логическую переменную bol и устанавливает ее значение равным false.

aver = 0;//Устанавливает значение переменной aver равным 0.

for (int i = 0; i < y; i++) {//Начинается два вложенных цикла для перебора всех элементов массива.

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

if (array[j][i] >= 0) {//Проверяет, если текущий элемент массива больше или равен 0, устанавливает переменную bol в значение true и выходит из цикла.

bol = true;

break;

}

}

if (!bol) {//Если все элементы в столбце отрицательны, вычисляет сумму элементов столбца и выходит из цикла.

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

aver += array[j][i];

}

break;

}

bol = false;// Снова устанавливает переменную bol в значение false.

}

aver = aver / x;// Вычисляет среднее арифметическое для суммы элементов столбца.

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

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

array[i][j] = array[i][j] - aver;//Вычитает среднее арифметическое из каждого элемента массива

}

}

return aver;

}

int main() {

SetConsoleOutputCP(1251);

int x = 3, y = 4, a = -10, b = 10;

double\*\* matrix = new double\* [x];//Выделяет память для двумерного массива размером x \* y.

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

matrix[i] = new double[y];

srand((unsigned)time(NULL));//Инициализирует генератор случайных чисел текущим временем.

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

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

matrix[i][j] = rand() % (b - a) + a;

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

}

cout << '\n';

}

//Объявляет переменную aver и вызывает функцию average\_values для вычисления среднего значения элементов массива, передавая массив, его размеры и переменную aver по ссылке.

double aver = 0;

aver = average\_values(matrix, x, y, aver);

//Проверяет результат вычисления среднего значения и выводит соответствующее сообщение на экран

if (aver == 0) {

cout << "Нет" << endl;

}

else {

cout << "Среднее арифметическое: " << aver << endl;

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

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

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

}

}

}

//Освобождает выделенную память для массива.

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

delete[] matrix[i];

delete[] matrix;

return 0;

}

