Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ковровская государственная технологическая академия имени В. А. Дегтярёва»

Факультет Автоматики и Электроники

Кафедра Приборостроение

**МОДУЛЬ 2**

**Руководитель:** Травкин Д. Н.

**Исполнитель:** студент группы У-120

Мартынов Р.С.

Ковров 2022

**Задание к работе**

Изм.

Лист

№ докум.

Подп.

Дата

Лист

2

1. Ввести массив, состоящий из 15 элементов целого типа. Упорядочить массив так, чтобы все отрицательные числа были расположены вначале по возрастанию, а все положительные – в конце по убыванию.

#include <iostream>

using namespace std;

void logArray(int arr[], int length)

{

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

{

int el = arr[i];

cout << el << "\t";

}

cout << "=======================================" << endl;

}

int \*bubbleSort(int arr[], int length, bool isAscending = true)

{

int sorted[length];

for(int i = 1; i < length; i++)

{

for(int k = 1; k < length; k++)

{

int prev = arr[k - 1];

int current = arr[k];

if(isAscending ? current < prev : current > prev)

{

arr[k - 1] = current;

arr[k] = prev;

}

}

}

return arr;

}

int main()

{

const int LENGTH = 15;

int arr[LENGTH];

int lessThanZero[LENGTH];

int greaterThanZero[LENGTH];

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

{

cin >> arr[i];

}

int lessThanZeroLastIndex = 0;

int greaterThanZeroLastIndex = 0;

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

{

int el = arr[i];

int isLowerThanZero = el < 0;

int \*targetArray = isLowerThanZero ? lessThanZero : greaterThanZero;

int &targetIndex = isLowerThanZero ? lessThanZeroLastIndex : greaterThanZeroLastIndex;

targetArray[targetIndex++] = el;

}

int \*sortedLessThanZero = bubbleSort(lessThanZero, lessThanZeroLastIndex);

int \*sortedGreaterThanZero = bubbleSort(greaterThanZero, greaterThanZeroLastIndex, false);

int merged[lessThanZeroLastIndex + greaterThanZeroLastIndex];

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

{

merged[i] = sortedLessThanZero[i];

}

for(int i = lessThanZeroLastIndex, k = 0; i < LENGTH; i++, k++)

{

merged[i] = sortedGreaterThanZero[k];

}

logArray(merged, lessThanZeroLastIndex + greaterThanZeroLastIndex);

}

2. Дан массив А[3][3]. Найти строку, сумма элементов которой максимальна, и умножить на максимальную сумму каждый элемент матрицы. Вывести получившийся массив на экран.

#include <iostream>

using namespace std;

int main()

{

const int COLS =3;

const int ROWS = COLS;

int arr[ROWS][COLS] ={ {1,7,2} , {0,-2, 4} , {5,9,-8} };

int totalSum;

for(int i = 0; i < 3; i++){

int sum=0;

for(int j = 0; j < 3; j++){

sum += arr[i][j];

}

if(i == 0 || sum > totalSum){

totalSum = sum;

}

}

cout << "Максимальная сумма: " << totalSum << endl;

for(int i = 0; i < 3; i++){

for(int j = 0 ;j < 3; j++){

arr[i][j] \*= totalSum;

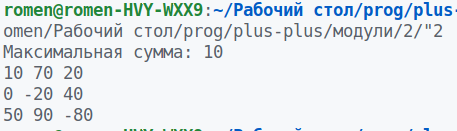
cout << arr[i][j] << " ";

}

cout << endl;

}

}



3. Дана матрица целых чисел размера NxM. Вывести номер строки, содержащей максимальное число одинаковых элементов.

#include <iostream>

#include <map>

using namespace std;

int main()

{

const int N=5;

const int M=5;

int arr[N][M]={{1,2,3,4,4},{5,0,1,1,5},{1,2,3,4,5},{1,2,4,5,6},{4,5,3,4,4}};

cout<<"Исходный массив: "<<endl;

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

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

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

}

cout<<endl;

}

int max=0;

int index=0;

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

map<int,int> numbers;

int localMax=1;

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

if(numbers.count(arr[i][j])){

numbers[arr[i][j]]++;

if(numbers[arr[i][j]]>localMax){

localMax=numbers[arr[i][j]];

}

}

else{

numbers.insert({arr[i][j],0});

}

}

if(localMax > max) {

max=localMax;

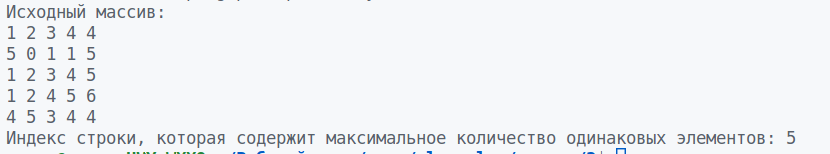
index=i;

}

}

cout<<"Индекс строки, которая содержит максимальное количество одинаковых элементов: "<<index+1<<endl;

}



4. Дана целочисленная квадратная матрица *{aij} n × n*. Написать программу, упорядочивания ее строк по возрастанию сумм их элементов.

#include <iostream>

using namespace std;

int main()

{

int N=10;

int \*\*arr = new int\* [N];

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

{

arr[i] = new int[N];

int rowSum = 0;

for(int k = 0; k < N; k++)

{

int el = arr[i][k] = rand() % 10;

rowSum += el;

cout << el << "\t";

}

cout << "\t" << "Sum - " << rowSum;

cout << endl;

}

for(int i = 1; i < N; i++)

{

for(int k = 1; k < N; k++)

{

int currentSum = 0;

int prevSum = 0;

int \*currentArr = arr[k];

int \*prevArr = arr[k - 1];

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

{

currentSum += currentArr[j];

prevSum += prevArr[j];

}

if(currentSum < prevSum)

{

arr[k - 1] = currentArr;

arr[k] = prevArr;

}

}

}

cout << "===============" << endl;

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

{

int rowSum = 0;

for(int k = 0; k < N; k++)

{

int el = arr[i][k];

rowSum += el;

cout << el << "\t";

}

cout << "\t" << "Sum - " << rowSum;

cout << endl;

}

delete[] arr;

}