Министерство образования и науки Российской Федерации

Севастопольский Государственный Университет

Кафедра ИС

ОТЧЕТ

по лабораторной работе №4

ОБРАБОТКА ДВУМЕРНЫХ МАССИВОВ

С ПОМОЩЬЮ ФУНКЦИЙ

Выполнил: ст. гр. ИТб-11

Куркчи А.Э.

Проверил:

Забаштанский А.К.

Севастополь

2015

# 1. Цель

Изучить основные принципы работы с массивами в языках С/С++. Исследовать способы передачи параметров в функции.

2. Постановка задач

Каждый пункт нижеприведенного задания оформить в виде функции. Все необходимые данные для функций должны передаваться им в качестве параметров. Ввод-вывод данных и результатов также организовать с помощью соответствующих функций.

Вариант №2

Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента. Характеристикой строки целочисленной матрицы назовём сумму её положительных чётных элементов. Переставляя строки заданной матрицы расположить их в соответствии с ростом характеристик.

3. Структурная схема

На рисунке 1 изображена схема алгоритма работы программы.

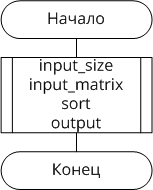
****

Рисунок 1 – Структурная схема

На рисунке 2 изображена схема алгоритма работы подпрограммы sort.

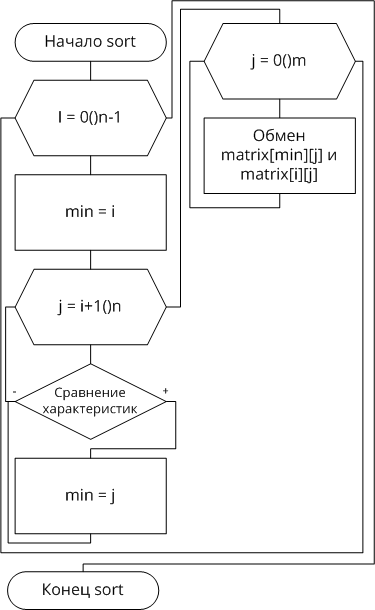
****

Рисунок 2 – Структурная схема sort

На рисунке 3 изображена схема алгоритма работы подпрограммы input\_size.

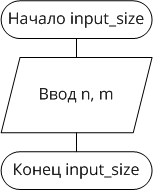
****

Рисунок 3 – Структурная схема input\_size

На рисунке 4 изображена схема алгоритма работы подпрограммы output.

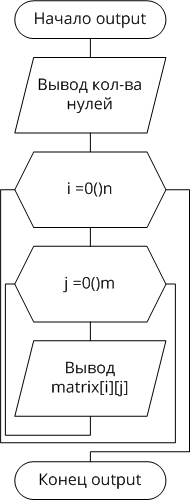
****

Рисунок 4 – Структурная схема output

На рисунке 5 изображена схема алгоритма работы подпрограммы count\_zero.

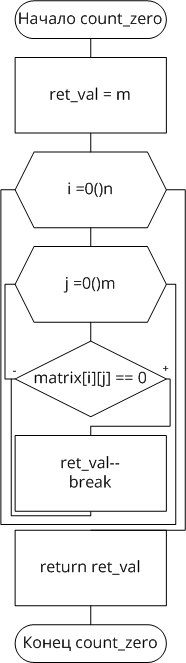
****

Рисунок 5 – Структурная схема count\_zero

На рисунке 6 изображена схема алгоритма работы подпрограммы input\_matrix.

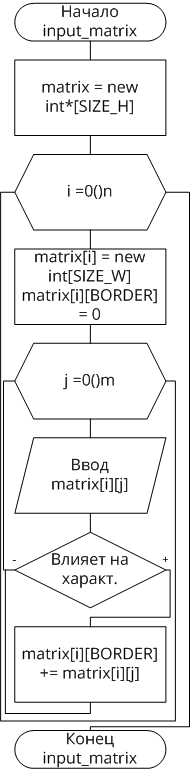
****

Рисунок 6 – Структурная схема input\_matrix

4. Текст программы

#include <iomanip>

#include <iostream>

#define SIZE\_W 101

#define SIZE\_H 101

#define BORDER 100

**using** **namespace** std**;**

// Ввод размеров матрицы

void input\_size**(**int**\*** n**,**int**\*** m**)** **{**

cout **<<** "Row count: "**;**

cin **>>** **\***n**;**

cout **<<** "Col count: "**;**

cin **>>** **\***m**;**

**}**

// Ввод матрицы

int **\*\*** input\_matrix**(**int**\*\*** matrix**,**int n**,**int m**)** **{**

cout **<<** "Elements :" **<<** endl**;**

matrix **=** **new** int**\*[**SIZE\_H**];**

matrix**[**BORDER**]** **=** **new** int**[**SIZE\_W**];**

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

matrix**[**BORDER**][**j**]** **=** 0**;**

**}**

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

matrix**[**i**]** **=** **new** int**[**SIZE\_W**];**

matrix**[**i**][**BORDER**]** **=** 0**;**

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

cin **>>** matrix**[**i**][**j**];**

**if((**matrix**[**i**][**j**]** **>** 0**)** **&&** **(**matrix**[**i**][**j**]** **%** 2 **==** 0**))** **{**

matrix**[**i**][**BORDER**]** **+=** matrix**[**i**][**j**];**

**}**

**}**

**}**

**return** matrix**;**

**}**

// Расчёт кол-ва нулей в столбцах

int count\_zero**(**int**\*\*** matrix**,**int n**,**int m**)** **{**

int ret\_val **=** m**;**

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

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

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

ret\_val**--;**

**break;**

**}**

**}**

**}**

**return** ret\_val**;**

**}**

// Сортировка матрицы

int**\*\*** sort**(**int**\*\*** matrix**,**int n**,**int m**)** **{**

int min**,**tmp**;**

**for(**int i**=**0**;**i**<**n**-**1**;**i**++)** **{**

min **=** i**;**

**for(**int j**=**i**+**1**;**j**<**n**;**j**++)** **{**

**if(**matrix**[**min**][**BORDER**]** **>** matrix**[**j**][**BORDER**])** **{**

min **=** j**;**

**}**

**}**

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

tmp **=** matrix**[**min**][**j**];**

matrix**[**min**][**j**]** **=** matrix**[**i**][**j**];**

matrix**[**i**][**j**]** **=** tmp**;**

**}**

**}**

**return** matrix**;**

**}**

// Вывод кол-ва столбцов без нулей и отсортированной матрицы

void output**(**int cnt**,**int**\*\*** matrix**,**int n**,**int m**)** **{**

cout **<<** "Cols without zero: " **<<** cnt **<<** endl**;**

cout **<<** "Sorted matrix:" **<<** endl**;**

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

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

cout **<<** setw**(**3**)** **<<** matrix**[**i**][**j**]** **<<** " "**;**

**}**

cout **<<** endl**;**

**}**

**}**

int main**(**int argc**,** char**\*\*** argv**)** **{**

int n**,**m**;**

int**\*\*** matrix**;**

input\_size**(&**n**,&**m**);**

matrix **=** input\_matrix**(**matrix**,**n**,**m**);**

matrix **=** sort**(**matrix**,**n**,**m**);**

output**(**count\_zero**(**matrix**,**n**,**m**),**matrix**,**n**,**m**);**

**delete** matrix**;**

**return** 0**;**

**}**

5. Тестовые примеры

На рисунке 7 представлены результаты работы программы

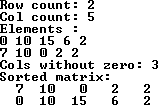


Рисунок 7 – Результат работы программы

Вывод

В ходе лабораторной работы были изучены основные принципы работы с массивами в языках C/C++. Исследованы способы передачи параметров в функции.