ВятГУ

Лабораторная работа №7,8

Вариант 13

ИНБс-1301-01-00 Харин Алексей

Проверил: Колупаев А.В.97

Киров 2021

**Цель работы:** Изучить правила работы с двумерными массивами. Получить навыки написания программ обработки двумерных массивов. Познакомиться с механизмом составления и организации взаимодействия пользовательских функций языка *С*.

**Текст задания:**

Дана целочисленная квадратная матрица:

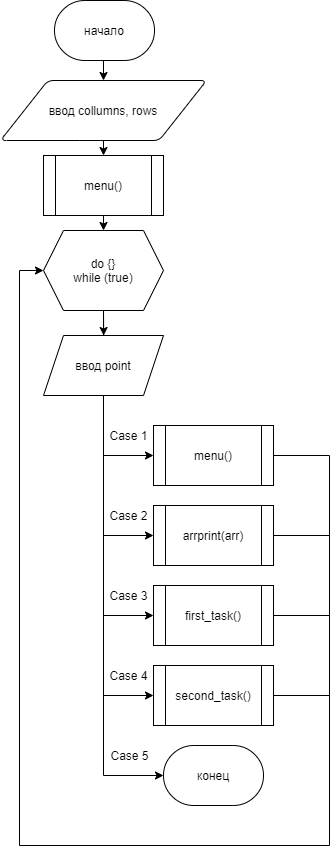
1. заменить нулями все элементы, кроме элементов, находящихся на главной диагонали;

2. в каждой строке поменять знак максимального по модулю элемента.

В функции main() организовать интерфейс взаимодействия с пользователем, посредством текстового меню; все операции ввода-вывода и вычислительные операции оформить как отдельные функции с параметрами и/или возвращаемым значением.

**Назначение идентификаторов (переменных):** collumns = 10 – количество столбцов (0 < collumns <= 100); rows = 8 – количество строк (0 < rows <= 100); arr[100][100] - массив (может содержать до 100 элементов строк и столбцов); point – указатель в меню; ftarr[100][100] – массив для первого задания (ftarr – first task array); starr[100][100] – массив для второго задания (starr – second task array); max\_el – максимальный элемент по модулю; max\_index – индекс максимального элемента по модулю в строке; i, j – элемент матрицы.

**Блок-схема алгоритма:**



|  |  |  |
| --- | --- | --- |
|  | |  |
|  |  | |

|  |  |
| --- | --- |
|  |  |

**Программные коды:**

// 2D array ver. 3.0 2021 Aleksey Kharin ©

#include <iostream>

#include <locale.h>

#include <math.h>

#include <string.h>

#include <typeinfo>

using namespace std;

double arr[100][100]; //Global arr parameter

int collumns, rows; //Global collumns and rows parameters

void foutput(const char str[256]) {

cout << str;

}

void arrgenerator() { //Array generetion

int i, j;

for (i = 0; i < rows; i++) {

for (j = 0; j < collumns; j++) {

arr[i][j] = rand() - rand();

}

}

}

void arrinput() { //Input array by kb

int i, j;

foutput("Введите массив поэлементно:\n");

for (i = 0; i < rows; i++) {

for (j = 0; j < collumns; j++) {

cout << "Элемент [" << i << "][" << j << "] = ";

cin >> arr[i][j];

}

}

}

void arrprint(double arr[100][100]) { //Output array

int i, j;

for (i = 0; i < rows; i++) {

foutput("[");

for (j = 0; j < collumns; j++) {

if (j != collumns - 1) { cout << "[" << arr[i][j] << "], "; }

else { cout << "[" << arr[i][j] << "]"; }

}

foutput("]\n");

}

}

void first\_task() { //First task

int i, j;

double ftarr[100][100];

for (i = 0; i < rows; i++) {

for (j = 0; j < collumns; j++) {

ftarr[i][j] = arr[i][j];

if (i != j) {

ftarr[i][j] = 0;

}

}

}

foutput("Array with \"0\" element instead of elements of main diagonal:\n");

arrprint(ftarr);

}

void second\_task() { //Second task

int i, j, max\_index;

double max\_el, starr[100][100];

for (i = 0; i < rows; i++) {

max\_el = 0;

for (j = 0; j < collumns; j++) {

starr[i][j] = arr[i][j];

if (max\_el <= abs(starr[i][j])) {

max\_el = starr[i][j];

max\_index = j;

}

}

if (starr[i][max\_index] > 0) { starr[i][max\_index] = 0 - starr[i][max\_index]; }

else { starr[i][max\_index] = abs(starr[i][max\_index]); }

}

foutput("Array with reversed max elements in a rows:\n");

arrprint(starr);

}

int menu\_1() { //Menu with editing array

int point;

foutput("Choose way to create array:\n"

"1. Input number of collumns and rows\n"

"2. Generate random\n"

"3. Input from keyboard\n"

"> ");

cin >> point;

return point;

}

int menu\_2() { //Output menu with tasks and exit

int point;

foutput("Programm menu:\n"

"1. Return to creating an array\n"

"2. Output array\n"

"3. First task\n"

"4. Second task\n"

"5. Exit programm\n"

"> ");

cin >> point;

return point;

}

int get\_num(const char str[256]) { //Inout collumns and rows

int x;

cout << str;

cin >> x;

if (x <= 0 || x > 100) {

foutput("Wrong input!");

exit(EXIT\_FAILURE);

}

return x;

}

void menu() { //Menu with tasks and exit

int point;

point = menu\_1();

switch (point) {

case 1:

collumns = get\_num("Input number of collumns (0 < collumns <= 100): ");

rows = get\_num("Input number of rows (0 < rows <= 100): ");

menu();

break;

case 2:

arrgenerator();

break;

case 3:

arrinput();

break;

default:

foutput("Wrong input!\n");

break;

}

}

int main() {

setlocale(LC\_ALL, "Russian");

srand((unsigned)time(0));

//declaring variables

int point;

//welcome screen

foutput("Лабортаторная работа №7,8\nвариант 13\nВыполнил: Харин Алексей\n");

collumns = get\_num("Input number of collumns (0 < collumns <= 100): ");

rows = get\_num("Input number of rows (0 < rows <= 100): ");

menu();

do { //Main menu

point = menu\_2();

switch (point) {

case 1:

menu();

break;

case 2:

foutput("Array is:\n");

arrprint(arr);

break;

case 3:

first\_task();

break;

case 4:

second\_task();

break;

case 5:

foutput("Programm completed.\n");

return 0;

default: //If input will be > 5

foutput("Wrong input!\n");

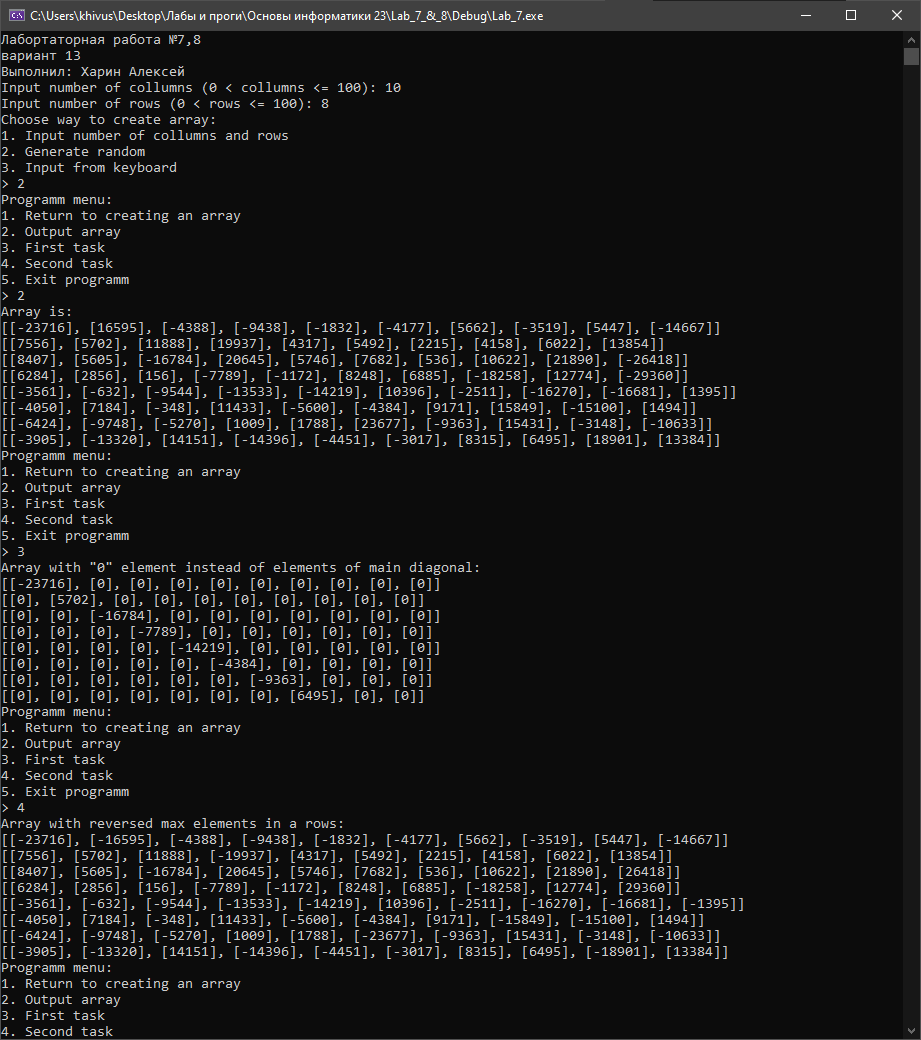
break;

}

} while (true);

}

**Экранные формы в процессе проектирования и выполнения:**



**Результаты вычисления:**

- 1 задание:

Матрица с 0 элементами, кроме основной диагонали:

[[-23716], [0], [0], [0], [0], [0], [0], [0], [0], [0]]

[[0], [5702], [0], [0], [0], [0], [0], [0], [0], [0]]

[[0], [0], [-16784], [0], [0], [0], [0], [0], [0], [0]]

[[0], [0], [0], [-7789], [0], [0], [0], [0], [0], [0]]

[[0], [0], [0], [0], [-14219], [0], [0], [0], [0], [0]]

[[0], [0], [0], [0], [0], [-4384], [0], [0], [0], [0]]

[[0], [0], [0], [0], [0], [0], [-9363], [0], [0], [0]]

[[0], [0], [0], [0], [0], [0], [0], [6495], [0], [0]]

- 2 задание:

Матрица с поменянным знаком у максимальных элементов в строчках:

[[-23716], [-16595], [-4388], [-9438], [-1832], [-4177], [5662], [-3519], [5447], [-14667]]

[[7556], [5702], [11888], [-19937], [4317], [5492], [2215], [4158], [6022], [13854]]

[[8407], [5605], [-16784], [20645], [5746], [7682], [536], [10622], [21890], [26418]]

[[6284], [2856], [156], [-7789], [-1172], [8248], [6885], [-18258], [12774], [29360]]

[[-3561], [-632], [-9544], [-13533], [-14219], [10396], [-2511], [-16270], [-16681], [-1395]]

[[-4050], [7184], [-348], [11433], [-5600], [-4384], [9171], [-15849], [-15100], [1494]]

[[-6424], [-9748], [-5270], [1009], [1788], [-23677], [-9363], [15431], [-3148], [-10633]]

[[-3905], [-13320], [14151], [-14396], [-4451], [-3017], [8315], [6495], [-18901], [13384]]