Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 4

«Многомерные массивы»

Проверил: Выполнил:

асс. каф. ЭВМ ст. гр. 250504

Скиба И.Г. Лагодич И.Р.

МИНСК 2022

Цель работы: научиться разрабатывать алгоритмы для работы с двумерными массивами и писать код на языке Си по составленному алгоритму. Лабораторная работа включает в себя 3 задачи для выполнения. Задачи решаются последовательно.

**Задание 1**

Условие: Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

* **Блок-схема алгоритма**





Рисунок 1.1 – Блок-схема алгоритма

* **Код**

#include <stdio.h>

#include <stdlib.h>

#include <ctime>

void manual\_input(int a[][100], int col, int row)

{

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

{

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

{

printf("Enter the [%d][%d] element of array: ", (i + 1), (j + 1));

while ((scanf\_s("%d", &a[i][j]) != 1) || (getchar() != '\n'))

{

printf("Wrong input. Enter the [%d][%d] element of array: ", (i + 1), (j + 1));

while (getchar() != '\n');

}

}

}

}

void output\_array(int a[][100], int col, int row)

{

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

{

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

{

printf("%d ", a[i][j]);

}

printf("\n");

}

}

void rand(int a[][100], int col, int row)

{

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

{

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

{

a[i][j] = rand() % 201 - 100;

}

}

}

void charact(int a[][100], int sum[], int col, int row)

{

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

{

sum[i] = 0;

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

{

if ((a[i][j] % 2 == 0) && (a[i][j] > 0))

sum[i] = sum[i] + a[i][j];

}

}

}

void output\_charact(int sum[], int col, int row)

{

printf("Charact array:\n");

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

{

printf("%d ", sum[i]);

printf("\n");

}

}

void main(void)

{

int a[100][100], sum[100], sort, col, row, choose, p;

srand(time(NULL));

printf("Enter amount of columns in the array: ");

while ((scanf\_s("%d", &col) != 1) || (col < 1 || col > 100) || (getchar() != '\n'))

{

printf("Wrong input. Enter amount of columns in the array: ");

while (getchar() != '\n');

}

printf("Enter amount of rows in the array: ");

while ((scanf\_s("%d", &row) != 1) || (row < 1 || row > 100) || (getchar() != '\n'))

{

printf("Wrong input. Enter amount of rows in the array: ");

while (getchar() != '\n');

}

printf("1.User filling\n2.Random filling\n");

printf("Choose the type of array filling: ");

while ((scanf\_s("%d", &choose) != 1) || (choose != 1 && choose != 2) || (getchar() != '\n'))

{

printf("Wrong input.\nChoose the type of array filling: ");

while (getchar() != '\n');

}

switch (choose)

{

case 1:

manual\_input(a, col, row);

break;

case 2:

rand(a, col, row);

break;

}

printf("Array:\n");

output\_array(a, col, row);

charact(a, sum, col, row);

//output\_charact(sum, col, row);

do

{

sort = 0;

for (int i = 0; i < col-1; i++)

{

if (sum[i] > sum[i + 1])

{

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

{

p = a[i][j];

a[i][j] = a[i + 1][j];

a[i + 1][j] = p;

}

sort = 1;

p = sum[i];

sum[i] = sum[i + 1];

sum[i + 1] = p;

}

}

} while (sort != 0);

printf("Changed array:\n");

output\_array(a, col, row);

}

* **Результат выполнения программы**



Рисунок 1.2 - Скриншот результата выполнения программы

**Задание 2**

Условие: Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер. Все элементы столбца с таким же номером уменьшить вдвое.

**2.1 Блок-схема алгоритма**





Рисунок 2.1 – Блок-схема алгоритма

**2.2 Исходный код**

#include <stdio.h>

#include <stdlib.h>

#include <ctime>

void manual\_input(int a[][100], int col, int row)

{

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

{

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

{

printf("Enter the [%d][%d] element of array: ", (i + 1), (j + 1));

while ((scanf\_s("%d", &a[i][j]) != 1) || (getchar() != '\n'))

{

printf("Wrong input. Enter the [%d][%d] element of array: ", (i + 1), (j + 1));

while (getchar() != '\n');

}

}

}

}

void output\_array(int a[][100], int col, int row)

{

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

{

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

{

printf("%d ", a[i][j]);

}

printf("\n");

}

}

void rand(int a[][100], int col, int row)

{

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

{

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

{

a[i][j] = rand() % 201 - 100;

}

}

}

void check(int a[][100], int col, int row, int& number)

{

number = 0;

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

{

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

{

if (a[i][j] < 0)

{

number = a[i][j];

int numberi = i;

int numberj = j;

printf("Number position: [%d][%d]\n", (numberi + 1), (numberj + 1));

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

{

a[i][j] = a[i][j] / 2;

}

goto label;

}

}

}

label:;

}

void main(void)

{

int a[100][100], col, row, choose, number;

srand(time(NULL));

printf("Enter amount of columns in the array: ");

while ((scanf\_s("%d", &col) != 1) || (col < 1 || col > 100) || (getchar() != '\n'))

{

printf("Wrong input. Enter amount of columns in the array: ");

while (getchar() != '\n');

}

printf("Enter amount of rows in the array: ");

while ((scanf\_s("%d", &row) != 1) || (row < 1 || row > 100) || (getchar() != '\n'))

{

printf("Wrong input. Enter amount of rows in the array: ");

while (getchar() != '\n');

}

printf("1.User filling\n2.Random filling\n");

printf("Choose the type of array filling: ");

while ((scanf\_s("%d", &choose) != 1) || (choose != 1 && choose != 2) || (getchar() != '\n'))

{

printf("Wrong input.\nChoose the type of array filling: ");

while (getchar() != '\n');

}

switch (choose)

{

case 1:

manual\_input(a, col, row);

break;

case 2:

rand(a, col, row);

break;

}

printf("Array:\n");

output\_array(a, col, row);

check(a, col, row, number);

if (number == 0)

printf("There is no negative element.");

else

{

printf("Changed array:\n");

output\_array(a, col, row);

}

}

**2.3 Результат выполнения программы**



Рисунок 2.2 - Скриншот результата выполнения программы

**Задание 3**



Рисунок 3.1 – Матрица разделённая на области

Условие: В квадратной матрице размером NxN найти минимальный элемент в 2-ой области

**3.1 Блок-схема алгоритма**





Рисунок 3.2 – Блок-схема алгоритма

**3.2 Исходный код**

#include <stdio.h>

#include <stdlib.h>

#include <ctime>

void manual\_input(int a[][100], int n)

{

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

{

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

{

printf("Enter the [%d][%d] element of array: ", (i + 1), (j + 1));

while ((scanf\_s("%d", &a[i][j]) != 1) || (getchar() != '\n'))

{

printf("Wrong input. Enter the [%d][%d] element of array: ", (i + 1), (j + 1));

while (getchar() != '\n');

}

}

}

}

void output\_array(int a[][100], int n)

{

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

{

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

{

printf("%d ", a[i][j]);

}

printf("\n");

}

}

void rand(int a[][100], int n)

{

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

{

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

{

a[i][j] = rand() % 201 - 100;

}

}

}

void find\_min(int a[][100], int n, int& min)

{

int ihalf, zn = 0;

min = a[0][n-1];

if (n % 2 == 0)

ihalf = n / 2;

if (n % 2 != 0)

ihalf = n / 2 + 1;

for (int j = n-1; j >= ihalf; j--)

{

for (int i = zn; i <= ihalf; i++)

{

if (min > a[i][j])

min = a[i][j];

if (min > a[(n-1) - i][j])

min = a[(n-1) - i][j];

zn++;

}

}

}

void main(void)

{

int a[100][100], n, choose, min;

srand(time(NULL));

printf("Enter the N: ");

while ((scanf\_s("%d", &n) != 1) || (n<1 || n>100) || (getchar() != '\n'))

{

printf("Wrong input. Enter the N: ");

while (getchar() != '\n');

}

printf("1.User filling\n2.Random filling\n");

printf("Choose the type of array filling: ");

while ((scanf\_s("%d", &choose) != 1) || (choose != 1 && choose != 2) || (getchar() != '\n'))

{

printf("Wrong input.\nChoose the type of array filling: ");

while (getchar() != '\n');

}

switch (choose)

{

case 1:

manual\_input(a, n);

break;

case 2:

rand(a, n);

break;

}

printf("Array:\n");

output\_array(a, n);

find\_min(a, n, min);

printf("Minimal element in the 2 area is: %d", min);

}

**3.3 Результат выполнения программы**



Рисунок 3.3 - Скриншот результата выполнения программы