**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №3**

**по дисциплине «Программирование»**

**Тема** : "**ДВУМЕРНЫЕ СТАТИЧЕСКИЕ МАССИВЫ. УКАЗАТЕЛИ"**

|  |  |  |
| --- | --- | --- |
| Студент гр. 2372 |  | Тубшинов В.Т. |
| Преподаватель |  | Глущенко А. Г. |

Санкт-Петербург

2022

**Цель работы.**

Изучение двумерных статических массив и основных алгоритмов работы с ними. Изучение ссылок и указателей, арифметики указателей. Написание программы, заполняющей матрицу размера N (N = 6, 8, 10) и заполняющий его паттернами «змейка» и «спираль». Программа должна так же менять местами блоки (подмассивы) матрицы в соответствии с четырьмя паттернами обмена из задания. Сортировать матрицу с использованием арифметики указателей и увеличивать, уменьшать, умножать и делить все элементы матрицы на число, введенное пользователем

**Основные теоретические положения.**

Указатели поддерживают ряд операций: присваивание, получение адреса указателя, получение значения по указателю, некоторые арифметические операции и операции сравнения.

К указателям можно применять некоторые арифметические операции. К таким операциям относятся :  **+**,**-**, **++**, **--**. Результаты выполнения этих операций по отношению к указателям существенно отличаются от результатов соответствующих арифметических операций, выполняющихся с обычными числовыми данными.

Добавлять к указателям или вычитать из указателей можно только целые значения.

Поскольку упомянутые арифметические операции выполняются по-разному при их применении к указателям и обычным арифметическим типам данных, а также учитывая высший приоритет операции \*, при использовании указателей в составе выражений следует внимательно обращаться со скобками.

Указатели – это очень мощное, полезное, но и очень опасное средство. Ошибки, которые возникают при неправильном использовании указателей, кроме того, что могут приводить к серьезным и непредсказуемым ошибкам в работе программы, еще и очень трудно диагностировать (обнаруживать).

**Постановка задач.**

1. Написать для каждого задания функцию, структурировать код.
2. Вывести двумерный массив двумя способами, используя работу с кареткой.
3. Отсортировать двумерный массив сортировкой «Вставками», не обращаясь к индексам элементов. (используя арифметику указателей)

**Выполнение работы.**

Код программы представлен в приложении А.

1. При запуске программы пользователь вводит размер квадратной матрицы.
2. Пользователю предлагается выбрать способ вывода массива.
3. Массив выводится змейкой или спиралью
4. Пользователю предлагается выбрать один из 4 методов перестановки блоков
5. Пользователю предлагается ввести число и операцию. Массив умножается/делится/уменьшается/увеличивается на введенное пользователем число.

**Выводы.**

Код написан, и работает корректно. Все задачи выполнены.

Приложение А

рабочий код

#include <iostream>

#include <windows.h>

#include <ctime>

using namespace std;

void PrintArray(int \*arr, int n) {

int \*p = arr;

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

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

cout << \*p << " ";

p++;

}

cout << endl;

}

}

int main() {

srand(time(0)); //для рандома

HANDLE hStdout; //ctrl + c , ctrl + v из дискорда

COORD destCoord;

hStdout = GetStdHandle(STD\_OUTPUT\_HANDLE);

const int n = 6; // размер массива

int arr[n][n]; // массив

int \*p = \*arr;; //указатель на массив

int k = 1; //счетчик

int move = 1; // направление движения 1 - вправо

int j = 0;

int i = 0; // индексы массива для вывода

int answer = 0; // repeat?

int task; // выбор задания

char variant; //для выбора a, b, c, d

//для второго задания

int temp[n][n];

//для третьего задания

int t = 0; // так как указатели не имеют оператора функции swap

int count = 0; // счетчик

//для четвертого задания

char action; // действие

int number; // число с которым будут действовать элементы массива

do {

cout << "choose task(1-4)" << endl;

cin >> task;

switch (task) {

case 1:

cout << "a or b?" << endl;

cin >> variant;

system("cls");

switch (variant) {

case 'a':

k = 1;

i = 0;

j = 0;

move = 1;

p = \*arr;

destCoord.X = j;

destCoord.Y = i;

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

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

arr[i][j] = 0; //заполняем массив нулями, чтобы поставить условие

}

}

while (k <= n \* n) {

if (k == n) { //для первого прохода, чтобы изменилось направление заполнения

move = n;

}

if (\*(p + move) == 0 || k == n \* n) { //условие: следующий элемент массива равен нулю и если к = 36, то выполнится

\*p = rand() % 36 + 1;

destCoord.X = 5 \* j;

destCoord.Y = 3 \* i;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*p << '\r';

cout.flush();

Sleep(100);

if (move == 1) {

j++; //меняем координату j для вывода

}

else if (move == n) {

i++; //так же и с i

}

else if (move == -1) {

j--;

}

else if (move == -n) {

i--;

}

p += move;

k++;

}

else { //если наше условие не выполняется, то меняем направление

switch (move) {

case 1:

move = n; // n - вниз

break;

case n:

move = -1; // -1 - влево

break;

case -1:

move = -n; // -n - вверх

break;

case -n:

move = 1; // ну и 1 - вправо

break;

default:

break;

}

}

}

cout << "\n\n\n\n\n\n\n" << endl;

break;

case 'b':

move = n; // так как у нас заполнение массива идет сверху вниз, то первое направление движения будет вниз

k = 1; // устанавливаем начальные значения для указателя, координат, счетчика

i = 0;

j = 0;

p = \*arr;

destCoord.X = j;

destCoord.Y = i;

while (k <= n \* n) {

\*p = rand()%36 + 1;

destCoord.X = 5 \* j;

destCoord.Y = 3 \* i;

SetConsoleCursorPosition(hStdout, destCoord);

cout << \*p << '\r';

Sleep(100);

if (move == n && k % n != 0) { // если изменение направления происходит только тогда, когда наш столбец из 6 символов заполняется

// то изменение будет проходить когда счетчик делится нацело на 6

i++;

p += move;

}

else if (move == -n && k % n != 0) {

i--;

p += move;

}

else {

move = -move;

j++;

p += 1;

}

k++;

}

cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n" << endl;

break;

default:

break;

}

break;

case 2: //второе задание

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

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

temp[i][j] = arr[i][j]; //создаем такой же массив как и искомый

}

}

cout << "Array:\n";

PrintArray(\*arr, n);

cout << "choose variant (a,b,c,d)\n";

cin >> variant;

switch (variant) {

case 'a':

for (i = 0; i < n / 2; i++) { //рассматриваем часть массива, которая в два раза меньше искомого и просто меняем значения на нужные нам

for (j = 0; j < n / 2; j++) {

arr[i][j + n / 2] = temp[i][j];

arr[i][j] = temp[i + n / 2][j];

arr[i + n / 2][j + n / 2] = temp[i][j + n / 2];

arr[i + n / 2][j] = temp[i + n / 2][j + n / 2];

}

}

break;

case 'b':

for (i = 0; i < n / 2; i++) {

for (j = 0; j < n / 2; j++) {

arr[i][j] = temp[i + n / 2][j + n / 2];

arr[i][j + n / 2] = temp[i + n / 2][j];

arr[i + n / 2][j] = temp[i][j + n / 2];

arr[i + n / 2][j + n / 2] = temp[i][j];

}

}

break;

case 'c':

for (i = 0; i < n / 2; i++) {

for (j = 0; j < n / 2; j++) {

arr[i][j] = temp[i + n / 2][j];

arr[i][j + n / 2] = temp[i + n / 2][j + n / 2];

arr[i + n / 2][j] = temp[i][j];

arr[i + n / 2][j + n / 2] = temp[i][j + n / 2];

}

}

break;

case 'd':

for (i = 0; i < n / 2; i++) {

for (j = 0; j < n / 2; j++) {

arr[i][j] = temp[i][j + n / 2];

arr[i][j + n / 2] = temp[i][j];

arr[i + n / 2][j] = temp[i + n / 2][j + n / 2];

arr[i + n / 2][j + n / 2] = temp[i + n / 2][j];

}

}

break;

}

PrintArray(\*arr, n);

break;

case 3:

cout << "Array:\n";

PrintArray(\*arr, n);

k = 0;

count = 0;

p = \*arr;

while (k < n\*n) {

while (count < n\*n - 1) { //двойной вайл, так как после первой итерации на место встает только один элемент

if (\*p > \*(p + 1)) {

t = \*p;

\*p = \*(p + 1);

\*(p + 1) = t;

}

count++;

p++;

}

k++;

p = \*arr;

count = 0;

}

cout << "Sorted array:\n";

PrintArray(\*arr, n);

break;

case 4: //проходимся по каждому элементу массива и делаем с ним, то, что выбрали

k = 0;

p = \*arr;

cout << "Array:\n";

PrintArray(\*arr, n);

cout << "Choose action (+, -, \*, /)\n";

cin >> action;

cout << "Choose number:\n";

cin >> number;

switch (action) {

case '+':

while (k < n\*n) {

\*p = \*p + number;

k++;

p++;

}

case '-':

while (k < n\*n) {

\*p = \*p - number;

k++;

p++;

}

case '\*':

while (k < n\*n) {

\*p = \*p \* number;

k++;

p++;

}

case '/':

while (k < n\*n) {

\*p = \*p / number;

k++;

p++;

}

default: break;

}

PrintArray(\*arr, n);

default:

break;

}

cout << "repeat? (0 - no, 1 - yes)" << endl;

cin >> answer;

system("cls");

} while (answer);

}