**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Информационных Систем и Технологий (ИСИТ)**

отчет

**по практической работе №3**

**по дисциплине «Программирование»**

Тема: «УКАЗАТЕЛИ»

|  |  |  |
| --- | --- | --- |
| Студент гр. XXXX |  | Стамбровский Т.С |
| Преподаватель |  | Глущенко А.Г. |

Санкт-Петербург

2020

**Цель работы.**

Ознакомление с основными особенностями работы с указателями в языке программирования C++, приобретение практических навыков работы с указателями, реализация различных алгоритмов, связанных с указателями.

**Основные теоретические положения.**

Указатели и ссылки являются одними из самых важных и достаточно сложных для понимания и использования средств языка программирования. Они ориентированы на прямую работу с памятью компьютера. С помощью этих средств реализуется работа с динамической памятью и динамическими объектами, возвращение из функций измененных данных и многое другое. К использованию указателей и ссылок мы будем неоднократно возвращаться в последующих разделах.

Все данные (переменные, константы и др.) хранятся в памяти. Память представляет собой непрерывную последовательность ячеек (байтов), каждая из которых имеет свой номер – адрес:

При определении, например, некоторой переменной, она располагается в памяти по определенному адресу и занимает столько ячеек, сколько требует тип этой переменной. Пусть, например, имеется переменные int A = 2351 и double B = 3.1 и пусть они располагаются в памяти так:

Говорят, что переменная А располагается по адресу 101 и занимает 4 байта, а переменная B имеет адрес 105 и занимает 8 байт памяти.

Для получения адреса какого-либо программного объекта используется оператор &. Например, если выполнить фрагмент следующей программы (в предположении, что переменные A и B располагаются в памяти, как это показано на предыдущем рисунке):

int A = 2351;

double B = 3.14;

cout << “Значение переменной А: ” << A << endl;

cout << “Адрес переменной А: ” << &A << endl;

cout << “Значение переменной В: ” << В << endl;

cout << “Адрес переменной В: ” << &В << endl;

получим следующий результат:

Значение переменной А: 2351

Адрес переменной А: 101

Значение переменной В: 3.14

Адрес переменной В: 105

Правда, значения адресов переменных будут выведены в шестнадцатеричном формате.

Указатели – это тоже обычные переменные, но они служат для хранения адресов памяти.

Указатели определяются в программе следующим образом:

<тип данных> \*<имя переменной>

Здесь <тип данных> определяет так называемый базовый тип указателя.

<Имя переменной> является идентификатором переменной-указателя.

Признаком того, что это переменная указатель, является символ \*, располагающийся между базовым типом указателя и именем переменной-указателя.

Например:

int \*p1;

double \*p2;

Здесь определены две переменные-указатели (или просто – два указателя). Указатель p1 является переменной-указателем на базовый тип int (или, как говорят, переменная p1 указывает на int - значение), а указатель p2 указывает на double – значение.

Иными словами, переменная p1 предназначена для хранения адресов участков памяти, размер которых соответствует типу int (4 байта), а переменная p2 - для хранения адресов участков памяти, размер которых соответствует типу double (8 байт).

Формально указатели представляют собой обычные целые значения типа int и занимают в памяти 4 байта не зависимо от базового типа указателя. Значения указателей при их выводе на экран представляются как целые значения в шестнадцатеричном формате.

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

Присваивание

Указателю можно присвоить либо адрес объекта того же типа, либо значение другого указателя.

Присвоение указателю адреса уже рассматривалось в прошлой теме. Для получения адреса объекта используется операция &:

int a = 1917;

int \*pa = &a; // указатель pa хранит адрес переменной a

При этом указатель и переменная должны иметь один и тот же тип, в данном случае это тип int.

Присвоение указателю другого указателя:

#include <iostream>

using std::cout;

using std::endl;

int main()

{

int a = 10;

int b = 2;

int \*pa = &a;

int \*pb = &b;

cout << "Variable a: address=" << pa << "\t value=" << \*pa << endl;

cout << "Variable b: address=" << pb << "\t value=" << \*pb << endl;

pa = pb; // теперь указатель pa хранит адрес переменной b

cout << "Variable b: address=" << pa << "\t value=" << \*pa << endl;

return 0;

}

Когда указателю присваивается другой указатель, то фактически первый указатель начинает также указывать на тот же адрес, на который указывает второй указатель.

Нулевые указатели

Нулевой указатель (null pointer) - это указатель, который не указывает ни на какой объект. Если мы не хотим, чтобы указатель указывал на какой-то конкретный адрес, то можно присвоить ему условное нулевое значение. Для создания нулевого указателя можно применять различные способы:

Ссылки на указатели

Так как ссылка не является объектом, то нельзя определить указатель на ссылку, однако можно определить ссылку на указатель. Через подобную ссылку можно изменять значение, на которое указывает указатель или изменять адрес самого указателя:

Разыменование указателя

Операция разыменования указателя представляет выражение в виде \*имя\_указателя. Эта операция позволяет получить объект по адресу, который хранится в указателе.

Через выражение \*pa мы можем получить значение по адресу, который хранится в указателе pa, а через выражение типа \*pa = значение вложить по этому адресу новое значение.

И так как в данном случае указатель pa указывает на переменную a, то при изменении значения по адресу, на который указывает указатель, также изменится и значение переменной a.

Адрес указателя

Указатель хранит адрес переменной, и по этому адресу мы можем получить значение этой переменной. Но кроме того, указатель, как и любая переменная, сам имеет адрес, по которому он располагается в памяти. Этот адрес можно получить также через операцию &:

Операции сравнения

К указателям могут применяться операции сравнения >, >=, <, <=,==, !=. Операции сравнения применяются только к указателям одного типа и к значениям NULL и nullptr. Для сравнения используются номера адресов:

Приведение типов

Иногда требуется присвоить указателю одного типа значение указателя другого типа. В этом случае следует выполнить операцию приведения типов с помощью операции (тип\_указателя \*):

Для преобразования указателя к другому типу в скобках перед указателем ставится тип, к которому надо преобразовать. Причем если мы не можем просто создать объект, например, переменную типа void, то для указателя это вполне будет работать. То есть можно создать указатель типа void.

Кроме того, следует отметить, что указатель на тип char (char \*pc = &c) при выводе на консоль система интерпретирует как строку

std::cout << "pc=" << pc << std::endl;

Поэтому если мы все-таки хотим вывести на консоль адрес, который хранится в указателе типа char, то это указатель надо преобразовать к другому типу, например, к void\* или к int\*.

Арифметика указателей

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.

Рассмотрим следующий пример:

int A = 20, B = 30;

int \*p1 = &A;

Пусть переменные A и B расположены в памяти, например, так, как это показано на следующем рисунке:

Указатель p1 содержит адрес переменной A, который равен 100 и \*p1 будет равно значению переменной A, то есть 20. Выполним следующую операцию:

p1 = p1 + 1;

или, что то же самое:

p1++;

Значение указателя изменится и станет равным 104, а не 101, как, наверное, ожидалось. То есть теперь указатель ссылается уже на переменную B и значение \*p1 будет равно 30.

Таким образом, добавление или вычитание 1 из указателя приводит к изменению его значения на размер базового типа указателя. В общем случае, например, при выполнении следующей операции:

p1 = p1 + N; // N – некоторое целое значение

значение указателя увеличится на sizeof(<базовый тип указателя>) \* N и в нашем случае это приращение будет равно sizeof(int) \* N = 4 \* N. Так, если N = 4, а p1 = 100 , то значение указателя p1 увеличится на 16 и станет равно 116, и указатель будет ссылаться на данные, расположенные по адресу 116.

Внимание. 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. Например, выражения (см. предыдущий рисунок)

\*(p1 + 1) и \*p1 + 1

имеют совершенно разный смысл. Первое выражение даст значение 30, а второе выражение будет равно 21 (в первом выражении сначала изменяется адрес, а затем осуществляется обращение в память по этому измененному адресу; во втором выражении мы обращаемся по старому адресу и к значению, хранящемуся по этому адресу добавляем 1).

**Постановка задачи.**

Необходимо написать программу, которая:

1) Используя арифметику указателей, заполняет квадратичную целочисленную матрицу порядка N (6,8,10) случайными числами от 1 до N\*N согласно схемам, приведенным на рисунках. Пользователь должен видеть процесс заполнения квадратичной матрицы.



2) Получает новую матрицу, из матрицы п. 1, переставляя ее блоки в соответствии со схемами:



3) Используя арифметику указателей, сортирует элементы любой сортировкой.

4) Уменьшает, увеличивает, умножает или делит все элементы матрицы на введенное пользователем число.

**Выполнение работы.**

Поочередно выполняем поставленные задачи, применяя на практике знания об указателях и арифметике указателей, во время работы с первым пунктом программы, проделываем необходимые манипуляции с кареткой, заполняя матрицу согласно схемам. Форматируем, добавляем цвет, создаем контекстное меню, внося в него пункты выполненных задач при помощи конструкции switch.

**Выводы.**

Были освоены новые приемы работы с указателями, изучены незнакомые способы работы с матрицами.

Приложение А

Полный код программы

#include <iostream>

#include <ctime>

#include <iomanip>

#include <windows.h>

using namespace std;

void wait(double seconds)

{

clock\_t endwait;

endwait = clock() + seconds \* CLOCKS\_PER\_SEC;

while (clock() < endwait) {}

}

void gotoxy(int x, int y)

{

HANDLE hConsole;

hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

COORD coord;

coord.X = x;

coord.Y = y;

SetConsoleCursorPosition(hConsole, coord);

}

int main()

{

HANDLE N = GetStdHandle(STD\_OUTPUT\_HANDLE);

setlocale(LC\_ALL, "Russian");

srand(time(NULL));;

int answer, answer\_1, answer\_2;

answer = 0;

answer\_2 = 0;

answer\_1 = 0;

while (answer\_1 != 9) {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите задание: " << endl;

cout << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "1. Заполнить матрицу по спирали" << endl;

cout << "2. Заполнить матрицу 'змейкой'" << endl;

cout << "3. Переставить блоки матрицы (пункт а)'" << endl;

cout << "4. Переставить блоки матрицы (пункт б)'" << endl;

cout << "5. Переставить блоки матрицы (пункт в)'" << endl;

cout << "6. Переставить блоки матрицы (пункт г)'" << endl;

cout << "7. Сортировать элементы матрицы " << endl;

cout << "8. Уменьшить / увеличить / умножить / разделить элементы матрицы на введенное число " << endl;

cout << "9. ";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout <<"Выход " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer\_1;

system("CLS");

switch (answer\_1)

{

case (1): {

do {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите размер матрицы: " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

cout << "1. Матрица 6x6" << endl;

cout << "2. Матрица 8x8" << endl;

cout << "3. Матрица 10x10" << endl;

cout << "4.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer;

system("CLS");

switch (answer)

{

case (1): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

int \*ptr = &arr[0][0];

////////////////////////////////// 1

gotoxy(5, 10);

for (int i = 0; i < 5; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int cc = 10;

for (int i = 5; i < 10; i++)

{

gotoxy(20, cc);

cc++;

cout << ptr[i] << " ";

wait(0.1);

}

int cc1 = 19;

for (int i = 10; i < 15; i++)

{

gotoxy(cc1, 15);

cc1--;

cc1--;

cc1--;

cout << " " << ptr[i];

wait(0.1);

}

int cc2 = 15;

for (int i = 15; i < 20; i++) {

gotoxy(5, cc2);

cc2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 2

gotoxy(8, 11);

for (int i = 20; i < 24; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

gotoxy(40, 20);

int ccc = 12;

for (int i = 24; i < 27; i++)

{

gotoxy(17, ccc);

ccc++;

cout << ptr[i] << endl;

wait(0.1);

}

int ccc1 = 13;

for (int i = 27; i < 30; i++)

{

gotoxy(ccc1, 14);

ccc1--;

ccc1--;

ccc1--;

cout << " " << ptr[i];

wait(0.1);

}

int ccc2 = 13;

for (int i = 30; i < 32; i++) {

gotoxy(8, ccc2);

ccc2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 3

gotoxy(11, 12);

for (int i = 32; i < 34; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int cccc = 13;

for (int i = 34; i < 35; i++)

{

gotoxy(14, cccc);

cccc++;

cout << ptr[i] << endl;

wait(0.1);

}

int сссс1 = 10;

for (int i = 35; i < 36; i++)

{

gotoxy(сссс1, 13);

сссс1--;

сссс1--;

сссс1--;

cout << " " << ptr[i];

wait(0.1);

}

gotoxy(0, 0);

break;

}

case (2): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

int \*ptr = &arr[0][0];

////////////////////////////////// 1

gotoxy(5, 10);

for (int i = 0; i < 8; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int cc = 11;

for (int i = 8; i < 15; i++)

{

gotoxy(26, cc);

cc++;

cout << ptr[i] << " ";

wait(0.1);

}

int cc1 = 22;

for (int i = 15; i < 21; i++)

{

gotoxy(cc1, 17);

cc1--;

cc1--;

cc1--;

cout << " " << ptr[i];

wait(0.1);

}

int cc2 = 17;

for (int i = 21; i < 28; i++) {

gotoxy(5, cc2);

cc2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 2

gotoxy(8, 11);

for (int i = 28; i < 34; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

gotoxy(40, 20);

int ccc = 12;

for (int i = 34; i < 39; i++)

{

gotoxy(23, ccc);

ccc++;

cout << ptr[i] << endl;

wait(0.1);

}

int ccc1 = 19;

for (int i = 39; i < 43; i++)

{

gotoxy(ccc1, 16);

ccc1--;

ccc1--;

ccc1--;

cout << " " << ptr[i];

wait(0.1);

}

int ccc2 = 16;

for (int i = 43; i < 48; i++) {

gotoxy(8, ccc2);

ccc2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 3

gotoxy(11, 12);

for (int i = 48; i < 51; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int cccc = 12;

for (int i = 51; i < 55; i++)

{

gotoxy(20, cccc);

cccc++;

cout << ptr[i] << endl;

wait(0.1);

}

int сссс1 = 16;

for (int i = 55; i < 58; i++)

{

gotoxy(сссс1, 15);

сссс1--;

сссс1--;

сссс1--;

cout << " " << ptr[i];

wait(0.1);

}

int cccc2 = 14;

for (int i = 58; i < 60; i++) {

gotoxy(11, cccc2);

cccc2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 4

gotoxy(14, 13);

for (int i = 60; i < 62; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int cccc3 = 14;

for (int i = 62; i < 63; i++)

{

gotoxy(17, cccc3);

cccc3++;

cout << ptr[i] << endl;

wait(0.1);

}

int сссс4 = 13;

for (int i = 63; i < 64; i++)

{

gotoxy(сссс4, 14);

сссс4--;

сссс4--;

сссс4--;

cout << " " << ptr[i];

wait(0.1);

}

gotoxy(0, 0);

break;

}

case (3): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

int \*ptr = &arr[0][0];

////////////////////////////////// 1

gotoxy(5, 10);

for (int i = 0; i < 10; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int ccg = 11;

for (int i = 10; i < 19; i++)

{

gotoxy(32, ccg);

ccg++;

cout << ptr[i] << " ";

wait(0.1);

}

int ccg1 = 28;

for (int i = 19; i < 28; i++)

{

gotoxy(ccg1, 19);

ccg1--;

ccg1--;

ccg1--;

cout << " " << ptr[i];

wait(0.1);

}

int ccg2 = 18;

for (int i = 28; i < 36; i++) {

gotoxy(5, ccg2);

ccg2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 2

gotoxy(8, 11);

for (int i = 36; i < 44; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int cc = 12;

for (int i = 44; i < 51; i++)

{

gotoxy(29, cc);

cc++;

cout << ptr[i] << " ";

wait(0.1);

}

int cc1 = 25;

for (int i = 51; i < 57; i++)

{

gotoxy(cc1, 18);

cc1--;

cc1--;

cc1--;

cout << " " << ptr[i];

wait(0.1);

}

int cc2 = 18;

for (int i = 57; i < 64; i++) {

gotoxy(8, cc2);

cc2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 3

gotoxy(11, 12);

for (int i = 64; i < 70; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int ccc = 13;

for (int i = 70; i < 75; i++)

{

gotoxy(26, ccc);

ccc++;

cout << ptr[i] << endl;

wait(0.1);

}

int ccc1 = 22;

for (int i = 75; i < 79; i++)

{

gotoxy(ccc1, 17);

ccc1--;

ccc1--;

ccc1--;

cout << " " << ptr[i];

wait(0.1);

}

int ccc2 = 17;

for (int i = 79; i < 84; i++) {

gotoxy(11, ccc2);

ccc2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 4

gotoxy(14, 13);

for (int i = 84; i < 87; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int cccc = 13;

for (int i = 87; i < 91; i++)

{

gotoxy(23, cccc);

cccc++;

cout << ptr[i] << endl;

wait(0.1);

}

int сссс1 = 19;

for (int i = 91; i < 94; i++)

{

gotoxy(сссс1, 16);

сссс1--;

сссс1--;

сссс1--;

cout << " " << ptr[i];

wait(0.1);

}

int cccc2 = 15;

for (int i = 94; i < 96; i++) {

gotoxy(14, cccc2);

cccc2--;

cout << ptr[i] << endl;

wait(0.1);

}

////////////////////////////////// 5

gotoxy(17, 14);

for (int i = 96; i < 98; i++)

{

cout << ptr[i] << " ";

wait(0.1);

}

int cccc3 = 15;

for (int i = 98; i < 99; i++)

{

gotoxy(20, cccc3);

cccc3++;

cout << ptr[i] << endl;

wait(0.1);

}

int сссс4 = 16;

for (int i = 99; i < 100; i++)

{

gotoxy(сссс4, 15);

сссс4--;

сссс4--;

сссс4--;

cout << " " << ptr[i];

wait(0.1);

}

gotoxy(0, 0);

break;

}

}

}

while (answer != 4);

break;

}

case(2): {

do {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите размер матрицы: " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

cout << "1. Матрица 6x6" << endl;

cout << "2. Матрица 8x8" << endl;

cout << "3. Матрица 10x10" << endl;

cout << "4.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer;

system("CLS");

switch (answer)

{

case (1): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

int h = 0;

int i1 = 0;

int p1 = 20;

int p2 = 23;

while (h != 36) {

int \*ptr = &arr[0][0];

gotoxy(5, 10);

int cc = 10;

for (int i = 0 + i1; i < 6 + i1; i++)

{

gotoxy(p1, cc);

cc++;

cout << ptr[i] << " ";

wait(0.1);

h++;

}

int cc1 = 15;

for (int i = 6 + i1; i < 12 + i1; i++) {

gotoxy(p2, cc1);

cc1--;

cout << ptr[i] << endl;

wait(0.1);

h++;

}

i1 += 12;

p1 += 6;

p2 += 6;

}

gotoxy(0, 0);

break;

}

case (2): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

int \*ptr = &arr[0][0];

int h = 0;

int i1 = 0;

int p1 = 20;

int p2 = 23;

while (h != 64) {

int \*ptr = &arr[0][0];

gotoxy(5, 10);

int cc = 10;

for (int i = 0 + i1; i < 8 + i1; i++)

{

gotoxy(p1, cc);

cc++;

cout << ptr[i] << " ";

wait(0.1);

h++;

}

int cc1 = 17;

for (int i = 8 + i1; i < 16 + i1; i++) {

gotoxy(p2, cc1);

cc1--;

cout << ptr[i] << endl;

wait(0.1);

h++;

}

i1 += 16;

p1 += 6;

p2 += 6;

}

gotoxy(0, 0);

break;

}

case (3): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

int \*ptr = &arr[0][0];

int h = 0;

int i1 = 0;

int p1 = 20;

int p2 = 23;

while (h != 100) {

int \*ptr = &arr[0][0];

gotoxy(5, 10);

int cc = 10;

for (int i = 0 + i1; i < 10 + i1; i++)

{

gotoxy(p1, cc);

cc++;

cout << ptr[i] << " ";

wait(0.1);

h++;

}

int cc1 = 19;

for (int i = 10 + i1; i < 20 + i1; i++) {

gotoxy(p2, cc1);

cc1--;

cout << ptr[i] << endl;

wait(0.1);

h++;

}

i1 += 20;

p1 += 6;

p2 += 6;

}

gotoxy(0, 0);

break;

}

}

}

while (answer != 4);

break;

}

case(3): {

do {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите размер матрицы: " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

cout << "1. Матрица 6x6" << endl;

cout << "2. Матрица 8x8" << endl;

cout << "3. Матрица 10x10" << endl;

cout << "4.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer;

system("CLS");

switch (answer)

{

case (1): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

cout << endl;

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

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

int temp = arr[i][j];

arr[i][j] = arr[i + 3][j];

arr[i + 3][j] = arr[i + 3][j + 3];

arr[i + 3][j + 3] = arr[i][j + 3];

arr[i][j + 3] = temp;

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

gotoxy(0, 15);

break;

}

case (2): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 4; ++i)

for (int j = 0; j < 4; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i + 4][j];

arr[i + 4][j] = arr[i + 4][j + 4];

arr[i + 4][j + 4] = arr[i][j + 4];

arr[i][j + 4] = temp;

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

gotoxy(0, 20);

break;

}

case (3): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 5; ++i)

for (int j = 0; j < 5; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i + 5][j];

arr[i + 5][j] = arr[i + 5][j + 5];

arr[i + 5][j + 5] = arr[i][j + 5];

arr[i][j + 5] = temp;

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

gotoxy(0, 22);

break;

}

}

}

while (answer != 4);

break;

}

case(4): {

do {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите размер матрицы: " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

cout << "1. Матрица 6x6" << endl;

cout << "2. Матрица 8x8" << endl;

cout << "3. Матрица 10x10" << endl;

cout << "4.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer;

system("CLS");

switch (answer)

{

case (1): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

cout << endl;

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

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

int temp = arr[i][j];

arr[i][j] = arr[i + 3][j + 3];

arr[i + 3][j + 3] = temp;

temp = arr[i + 3][j];

arr[i + 3][j] = arr[i][j + 3];

arr[i][j + 3] = temp;

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

gotoxy(0, 15);

break;

}

case (2): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 4; ++i)

for (int j = 0; j < 4; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i + 4][j + 4];

arr[i + 4][j + 4] = temp;

temp = arr[i + 4][j];

arr[i + 4][j] = arr[i][j + 4];

arr[i][j + 4] = temp;

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

gotoxy(0, 20);

break;

}

case (3): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 5; ++i)

for (int j = 0; j < 5; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i + 5][j + 5];

arr[i + 5][j + 5] = temp;

temp = arr[i + 5][j];

arr[i + 5][j] = arr[i][j + 5];

arr[i][j + 5] = temp;

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

gotoxy(0, 22);

break;

}

}

}

while (answer != 4);

break;

}

case(5): {

do {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите размер матрицы: " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

cout << "1. Матрица 6x6" << endl;

cout << "2. Матрица 8x8" << endl;

cout << "3. Матрица 10x10" << endl;

cout << "4.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer;

system("CLS");

switch (answer)

{

case (1): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

cout << endl;

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

for (int j = 0; j < 6; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i + 3][j];

arr[i+3][j] = temp;

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

gotoxy(0, 15);

break;

}

case (2): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 4; ++i)

for (int j = 0; j < 8; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i + 4][j];

arr[i + 4][j] = temp;

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

gotoxy(0, 20);

break;

}

case (3): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 5; ++i)

for (int j = 0; j < 10; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i + 5][j];

arr[i + 5][j] = temp;

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

gotoxy(0, 22);

break;

}

}

}

while (answer != 4);

break;

}

case(6): {

do {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите размер матрицы: " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

cout << "1. Матрица 6x6" << endl;

cout << "2. Матрица 8x8" << endl;

cout << "3. Матрица 10x10" << endl;

cout << "4.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer;

system("CLS");

switch (answer)

{

case (1): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 6; ++i)

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

int temp = arr[i][j];

arr[i][j] = arr[i][j+3];

arr[i][j+3] = temp;

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

gotoxy(0, 15);

break;

}

case (2): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 8; ++i)

for (int j = 0; j < 4; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i][j + 4];

arr[i][j + 4] = temp;

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

gotoxy(0, 20);

break;

}

case (3): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 10; ++i)

for (int j = 0; j < 5; ++j) {

int temp = arr[i][j];

arr[i][j] = arr[i][j + 5];

arr[i][j + 5] = temp;

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

gotoxy(0, 22);

break;

}

}

}

while (answer != 4);

break;

}

case(7): {

do {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите размер матрицы: " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

cout << "1. Матрица 6x6" << endl;

cout << "2. Матрица 8x8" << endl;

cout << "3. Матрица 10x10" << endl;

cout << "4.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer;

system("CLS");

switch (answer)

{

case (1): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

cout << \*(\*(arr + i) + j) << " ";

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 36; i++)

for (int j = 0; j < 35; j++)

if (\*(\*arr + j) > \*(\*arr + j + 1))

swap(\*(\*arr + j), \*(\*arr + j + 1));

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

cout << \*(\*(arr + i) + j) << " ";

}

cout << endl;

}

gotoxy(0, 15);

break;

}

case (2): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

cout << \*(\*(arr + i) + j) << " ";

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 64; i++)

for (int j = 0; j < 63; j++)

if (\*(\*arr + j) > \*(\*arr + j + 1))

swap(\*(\*arr + j), \*(\*arr + j + 1));

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

cout << \*(\*(arr + i) + j) << " ";

}

cout << endl;

}

gotoxy(0, 20);

break;

}

case (3): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

\*(\*(arr + i) + j) = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

cout << \*(\*(arr + i) + j) << " ";

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 100; i++)

for (int j = 0; j < 99; j++)

if (\*(\*arr + j) > \*(\*arr + j + 1))

swap(\*(\*arr + j), \*(\*arr + j + 1));

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

cout << \*(\*(arr + i) + j) << " ";

}

cout << endl;

}

gotoxy(0, 22);

break;

}

}

}

while (answer != 4);

break;

}

case(8): {

do {

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << "Выберите размер матрицы: " << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

cout << "1. Матрица 6x6" << endl;

cout << "2. Матрица 8x8" << endl;

cout << "3. Матрица 10x10" << endl;

cout << "4.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer;

system("CLS");

switch (answer)

{

case (1): {

int m;

cout << "Введите число: ";

SetConsoleTextAttribute(N, FOREGROUND\_BLUE | FOREGROUND\_INTENSITY);

cin >> m;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

do {

cout << "1. Уменьшить все элементы матрицы на введенное число " << endl;

cout << "2. Увеличить все элементы матрицы на введенное число " << endl;

cout << "3. Умножить все элементы матрицы на введенное число " << endl;

cout << "4. Разделить все элементы матрицы на введенное число " << endl;

cout << "5.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer\_2;

system("CLS");

switch (answer\_2)

{

case (1): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] -= m;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

gotoxy(0, 15);

break;

}

case (2): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] += m;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

gotoxy(0, 15);

break;

}

case (3): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

gotoxy(0, 15);

break;

}

case (4): {

int arr[6][6];

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

arr[i][j] /= m;

}

}

for (int i = 0; i < 6; i++) {

for (int j = 0; j < 6; j++) {

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

}

cout << endl;

}

gotoxy(0, 15);

break;

}

}

} while (answer\_2 != 5);

break;

}

case (2): {

int m;

cout << "Введите число: ";

SetConsoleTextAttribute(N, FOREGROUND\_BLUE | FOREGROUND\_INTENSITY);

cin >> m;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

do {

cout << "1. Уменьшить все элементы матрицы на введенное число " << endl;

cout << "2. Увеличить все элементы матрицы на введенное число " << endl;

cout << "3. Умножить все элементы матрицы на введенное число " << endl;

cout << "4. Разделить все элементы матрицы на введенное число " << endl;

cout << "5.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer\_2;

system("CLS");

switch (answer\_2)

{

case (1): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] -= m;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

gotoxy(0, 20);

break;

}

case (2): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] += m;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

gotoxy(0, 20);

break;

}

case (3): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

gotoxy(0, 20);

break;

}

case (4): {

int arr[8][8];

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

arr[i][j] /= m;

}

}

for (int i = 0; i < 8; i++) {

for (int j = 0; j < 8; j++) {

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

}

cout << endl;

}

gotoxy(0, 20);

break;

}

}

} while (answer\_2 != 5);

break;

}

case (3): {

int m;

cout << "Введите число: ";

SetConsoleTextAttribute(N, FOREGROUND\_BLUE | FOREGROUND\_INTENSITY);

cin >> m;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << endl;

do {

cout << "1. Уменьшить все элементы матрицы на введенное число " << endl;

cout << "2. Увеличить все элементы матрицы на введенное число " << endl;

cout << "3. Умножить все элементы матрицы на введенное число " << endl;

cout << "4. Разделить все элементы матрицы на введенное число " << endl;

cout << "5.";

SetConsoleTextAttribute(N, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cout << " <——" << endl;

SetConsoleTextAttribute(N, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);

cin >> answer\_2;

system("CLS");

switch (answer\_2)

{

case (1): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] -= m;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

gotoxy(0, 22);

break;

}

case (2): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] += m;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

gotoxy(0, 22);

break;

}

case (3): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

gotoxy(0, 22);

break;

}

case (4): {

int arr[10][10];

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] = 10 + rand() % 90;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

cout << endl;

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

arr[i][j] /= m;

}

}

for (int i = 0; i < 10; i++) {

for (int j = 0; j < 10; j++) {

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

}

cout << endl;

}

gotoxy(0, 22);

break;

}

}

} while (answer\_2 != 5);

break;

}

}

} while (answer != 4);

break;

}

break;

}

}

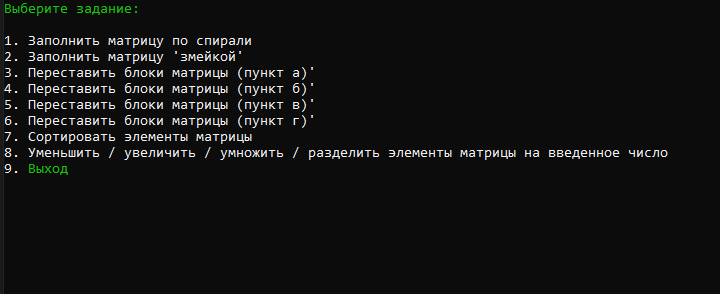
return 0;

}

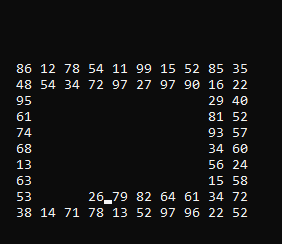
Приложение б

Полученные результаты

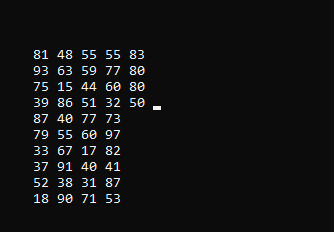
Меню:



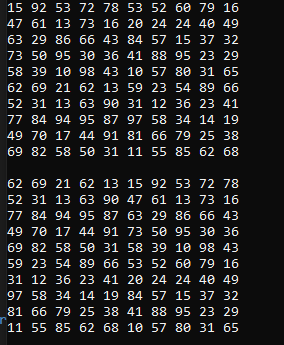
Процесс заполнения матрицы по спирали:

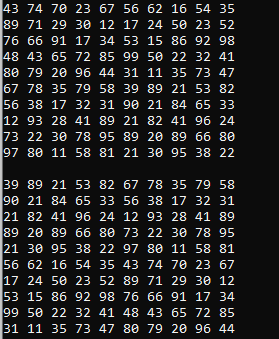


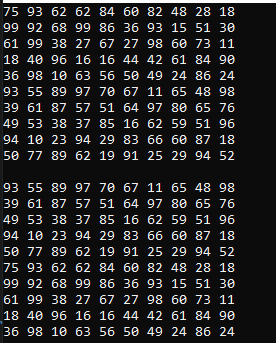
Процесс заполнения матрицы «змейкой»:

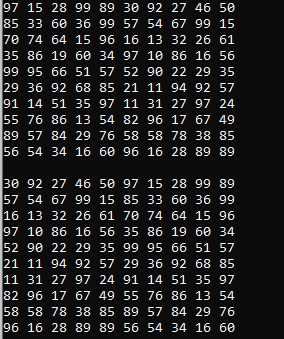


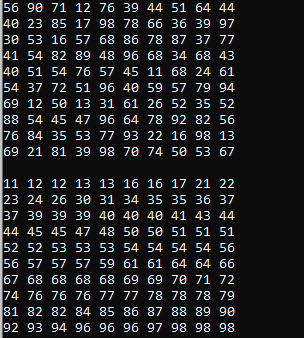
Перестановка элементов матрицы:

А) 

Б) 

В) 

Г) 

Сортировка матрицы: 

Пример изменения элементов матрицы (умножение на введенное число, равное 10): 