**онлайн-Платформа STEPIK**

**отчет**

**по практической работе №3**

**по курсу «Основы алгоритмизации и программирования на языке C++»**

**Тема: Двумерные статические массивы. Указатели**

|  |  |  |
| --- | --- | --- |
| Студент(ка) гр. | Простаков А.В (User ID: 89907427) |  |
| Преподаватель | Глущенко А.Г. |  |

Москва

2024

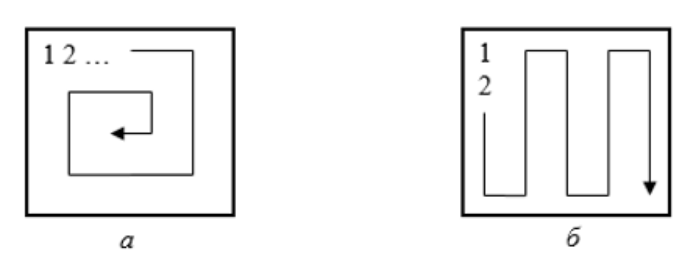
**Цель работы.**

Отработка практических навыков работы с указателями

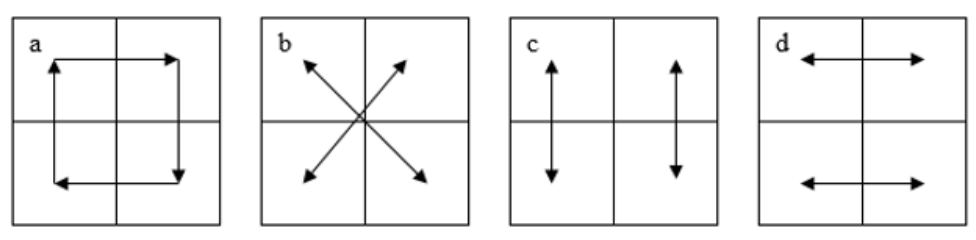
**Постановка задачи**

Разработать алгоритм и 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:



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



1. Используя арифметику указателей, сортирует элементы любой сортировкой из перечисленных: shaker sort, comb sort, insert sort, quick sort.
2. Уменьшает, увеличивает, умножает или делит все элементы матрицы на введенное пользователем число.

**Выполнение работы.**

Код программы в редактируемом формате представлен в приложении А.

**Выводы**

В данной практической работе рассмотрены приемы работы c двумерными статическими массивами с использованием арифметики указателей, то есть с прямым обращением к памяти компьютера.

**Приложение А**

**рабочий код**

|  |
| --- |
| 1. *Используя арифметику указателей, заполнить 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* |
| *Вывод массива в следующей последовательности:* |
| #include <iostream>  #include <iomanip>  #include <iostream>  #include <Windows.h>  #include <stdlib.h>  #include <time.h>  #include <chrono>  using namespace std;  int GetRand(int min, int max)  {  int num = min + rand() % (max - min + 1);  return num;  }  int main()  {  srand(time(NULL));  setlocale(LC\_ALL, "ru");  int arr[8][8];  int width = 3;  int k;  int\* first = nullptr;  int\* ptr = nullptr;  int\* aP = nullptr;  int n0 = sizeof(arr) / sizeof(\*arr);  int m0 = sizeof(\*arr) / sizeof(\*\*arr);  int\* end = \*arr + m0 \* n0 - 1;  int iter;  int n1, m1;  for (int\* ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = GetRand(1, 36);  }  HANDLE hStdout;  COORD destCoord;  hStdout = GetStdHandle(STD\_OUTPUT\_HANDLE);    for (aP = \*arr, n1 = n0, m1 = m0, iter = 0; n1 >= 2; aP = aP + (m0+1), n1 = n1 - 2, m1 = m1 - 2, iter++)  {  end = aP + (n1 - 1);  for (int\* ptr = aP, k = iter; ptr <= end; ptr++, k++)  {  destCoord.X = width \* k;  destCoord.Y = (ptr - \*arr) / m0;  SetConsoleCursorPosition(hStdout, destCoord);  cout<<\*ptr <<" ";  cout.flush();  Sleep(50);  }  first = end + n0;  end = first + n0 \* (m0 - 2 \* (iter + 1));  for (ptr = first; ptr <= end; ptr = ptr + n0)  {  k = m0 - iter - 1;  destCoord.X = width \* k;  destCoord.Y = (ptr - \*arr) / m0;  SetConsoleCursorPosition(hStdout, destCoord);  cout << \*ptr << " ";  cout.flush();  Sleep(50);  }  first = end - 1;  end = first - (n1 - 2);  for (ptr = first, k = (m0 - 2-iter); ptr >= end; ptr--, k--)  {  destCoord.X = width \* k;  destCoord.Y = (ptr - \*arr) /m0;  SetConsoleCursorPosition(hStdout, destCoord);  cout << \*ptr << " ";  cout.flush();  Sleep(50);  }  first = end - n0;  end = first - n0 \* (m0 - 2 \* (iter + 1));  for (ptr = first; ptr > end; ptr = ptr - m0)  {  destCoord.X = width \* iter;  destCoord.Y = (ptr - \*arr) /m0;  SetConsoleCursorPosition(hStdout, destCoord);  cout << \*ptr << " ";  cout.flush();  Sleep(50);  }  }    cout << "\n" << "\n" << "\n" << "\n" << "\n" << "\n" << "\n" << "\n" << "\n";    for (int l = 0; l < m0; l++)  {  for (int r = 0; r < n0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  } |
| *Вывод массива в следующей последовательности:* |
| #include <iostream>  #include <iomanip>  #include <iostream>  #include <Windows.h>  #include <stdlib.h>  #include <time.h>  #include <chrono>  using namespace std;  int GetRand(int min, int max)  {  int num = min + rand() % (max - min + 1);  return num;  }  int main()  {  srand(time(NULL));  setlocale(LC\_ALL, "ru");  int arr[8][8];  int width = 3;  int k;  int\* first = nullptr;  int\* ptr = nullptr;  int\* aP = nullptr;  int n0 = sizeof(arr) / sizeof(\*arr);  int m0 = sizeof(\*arr) / sizeof(\*\*arr);  int\* end = \*arr + m0 \* n0 - 1;  int iter;    for (int\* ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = GetRand(1, 36);  }  HANDLE hStdout;  COORD destCoord;  hStdout = GetStdHandle(STD\_OUTPUT\_HANDLE);    for (aP = \*arr, iter = 0, k=0; iter <= (n0 / 2 - 1); aP = aP + 2, iter++, k=k+2)  {  end = aP + n0 \* (m0 - 1);  for (ptr = aP; ptr <= end; ptr = ptr + n0)  {  destCoord.X = width \* k;  destCoord.Y = (ptr - \*arr) / m0;  SetConsoleCursorPosition(hStdout, destCoord);  cout << \*ptr << " ";  cout.flush();  Sleep(50);  }  first = end + 1;  end = first - n0 \* (m0 - 1);  for (ptr = first; ptr >= end; ptr = ptr - n0)  {  destCoord.X = width \* (k+1);  destCoord.Y = (ptr - \*arr) / m0;  SetConsoleCursorPosition(hStdout, destCoord);  cout << \*ptr << " ";  cout.flush();  Sleep(50);  }  }  cout << "\n" << "\n" << "\n" << "\n" << "\n" << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < m0; l++)  {  for (int r = 0; r < n0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  } |
| *2)* Получить новую матрицу, из матрицы п.1, переставляя ее блоки в соответствии со схемами: |
|  |
| #include <iostream>  #include <iomanip>  #include <iostream>  #include <Windows.h>  #include <stdlib.h>  #include <time.h>  #include <chrono>  using namespace std;  int GetRand(int min, int max)  {  int num = min + rand() % (max - min + 1);  return num;  }  int main()  {  srand(time(NULL));  setlocale(LC\_ALL, "ru");  const int M = 6;  const int N = 3;  int arr[M][M];  int arr1[N][N];  int arr2[N][N];  int arr3[N][N];  int arr4[N][N];  int width = 3;  int k;  int\* first = nullptr;  int\* ptr = nullptr;  int\* ptr1 = nullptr;  int\* aP = nullptr;  int\* aP1 = nullptr;  int n0 = sizeof(arr) / sizeof(\*arr);  int m0 = sizeof(\*arr) / sizeof(\*\*arr);  int\* end = \*arr + m0 \* n0 - 1;  int\* end1 = nullptr;  int iter;  int n1, m1;  for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = GetRand(1, 36);  }    cout << "Исходный вид массива" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  //Сборка первой коробки  for (aP = \*arr, end = \*arr + n0 / 2 - 1, aP1 = \*arr1, k = 1; k <= n0/2; aP = aP + n0, end = end+n0, aP1=aP1+(n0/2), k++)  {    for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка второй коробки  for (aP = \*arr+n0/2, end = \*arr + n0 - 1, aP1 = \*arr2, k = 1; k <= n0 /2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка третьей коробки  for (aP = \*arr + n0\*n0/2, end = \*arr +n0\*n0/2+n0/2-1, aP1 = \*arr3, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка четвертой коробки  for (aP = \*arr + n0 \* n0 / 2+n0/2, end = \*arr + n0 \* n0 / 2 + n0 / 2 + n0 / 2 - 1, aP1 = \*arr4, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }    //Запись третьей коробки  for (aP = \*arr, end = \*arr + n0 / 2 - 1, aP1 = \*arr3, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись первой коробки  for (aP = \*arr + n0 / 2, end = \*arr + n0 - 1, aP1 = \*arr1, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись третьей коробки  for (aP = \*arr + n0 \* n0 / 2, end = \*arr + n0 \* n0 / 2 + n0 / 2 - 1, aP1 = \*arr4, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись четвертой коробки  for (aP = \*arr + n0 \* n0 / 2 + n0 / 2, end = \*arr + n0 \* n0 / 2 + n0 / 2 + n0 / 2 - 1, aP1 = \*arr2, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }    for (int l = 0; l < n0/2; l++)  {  for (int r = 0; r < n0/2; r++)  {  cout << arr1[l][r] << " ";  if (r == (n0/2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0/2; l++)  {  for (int r = 0; r < n0/2; r++)  {  cout << arr2[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0 / 2; l++)  {  for (int r = 0; r < n0 / 2; r++)  {  cout << arr3[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0 / 2; l++)  {  for (int r = 0; r < n0 / 2; r++)  {  cout << arr4[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  cout << "Вид массива после перестановки" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  } |
|  |
| #include <iostream>  #include <iomanip>  #include <iostream>  #include <Windows.h>  #include <stdlib.h>  #include <time.h>  #include <chrono>  using namespace std;  int GetRand(int min, int max)  {  int num = min + rand() % (max - min + 1);  return num;  }  int main()  {  srand(time(NULL));  setlocale(LC\_ALL, "ru");  const int M = 6;  const int N = 3;  int arr[M][M];  int arr1[N][N];  int arr2[N][N];  int arr3[N][N];  int arr4[N][N];  int width = 3;  int k;  int\* first = nullptr;  int\* ptr = nullptr;  int\* ptr1 = nullptr;  int\* aP = nullptr;  int\* aP1 = nullptr;  int n0 = sizeof(arr) / sizeof(\*arr);  int m0 = sizeof(\*arr) / sizeof(\*\*arr);  int\* end = \*arr + m0 \* n0 - 1;  int\* end1 = nullptr;  int iter;  int n1, m1;  for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = GetRand(1, 36);  }    cout << "Исходный вид массива" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  //Сборка первой коробки  for (aP = \*arr, end = \*arr + n0 / 2 - 1, aP1 = \*arr1, k = 1; k <= n0/2; aP = aP + n0, end = end+n0, aP1=aP1+(n0/2), k++)  {    for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка второй коробки  for (aP = \*arr+n0/2, end = \*arr + n0 - 1, aP1 = \*arr2, k = 1; k <= n0 /2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка третьей коробки  for (aP = \*arr + n0\*n0/2, end = \*arr +n0\*n0/2+n0/2-1, aP1 = \*arr3, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка четвертой коробки  for (aP = \*arr + n0 \* n0 / 2+n0/2, end = \*arr + n0 \* n0 / 2 + n0 / 2 + n0 / 2 - 1, aP1 = \*arr4, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }    //Запись первой коробки  for (aP = \*arr, end = \*arr + n0 / 2 - 1, aP1 = \*arr4, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись второй коробки  for (aP = \*arr + n0 / 2, end = \*arr + n0 - 1, aP1 = \*arr3, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись третьей коробки  for (aP = \*arr + n0 \* n0 / 2, end = \*arr + n0 \* n0 / 2 + n0 / 2 - 1, aP1 = \*arr2, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись четвертой коробки  for (aP = \*arr + n0 \* n0 / 2 + n0 / 2, end = \*arr + n0 \* n0 / 2 + n0 / 2 + n0 / 2 - 1, aP1 = \*arr1, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }    for (int l = 0; l < n0/2; l++)  {  for (int r = 0; r < n0/2; r++)  {  cout << arr1[l][r] << " ";  if (r == (n0/2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0/2; l++)  {  for (int r = 0; r < n0/2; r++)  {  cout << arr2[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0 / 2; l++)  {  for (int r = 0; r < n0 / 2; r++)  {  cout << arr3[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0 / 2; l++)  {  for (int r = 0; r < n0 / 2; r++)  {  cout << arr4[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  cout << "Вид массива после перестановки" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }      } |
|  |
| #include <iostream>  #include <iomanip>  #include <iostream>  #include <Windows.h>  #include <stdlib.h>  #include <time.h>  #include <chrono>  using namespace std;  int GetRand(int min, int max)  {  int num = min + rand() % (max - min + 1);  return num;  }  int main()  {  srand(time(NULL));  setlocale(LC\_ALL, "ru");  const int M = 6;  const int N = 3;  int arr[M][M];  int arr1[N][N];  int arr2[N][N];  int arr3[N][N];  int arr4[N][N];  int width = 3;  int k;  int\* first = nullptr;  int\* ptr = nullptr;  int\* ptr1 = nullptr;  int\* aP = nullptr;  int\* aP1 = nullptr;  int n0 = sizeof(arr) / sizeof(\*arr);  int m0 = sizeof(\*arr) / sizeof(\*\*arr);  int\* end = \*arr + m0 \* n0 - 1;  int\* end1 = nullptr;  int iter;  int n1, m1;  for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = GetRand(1, 36);  }    cout << "Исходный вид массива" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  //Сборка первой коробки  for (aP = \*arr, end = \*arr + n0 / 2 - 1, aP1 = \*arr1, k = 1; k <= n0/2; aP = aP + n0, end = end+n0, aP1=aP1+(n0/2), k++)  {    for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка второй коробки  for (aP = \*arr+n0/2, end = \*arr + n0 - 1, aP1 = \*arr2, k = 1; k <= n0 /2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка третьей коробки  for (aP = \*arr + n0\*n0/2, end = \*arr +n0\*n0/2+n0/2-1, aP1 = \*arr3, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка четвертой коробки  for (aP = \*arr + n0 \* n0 / 2+n0/2, end = \*arr + n0 \* n0 / 2 + n0 / 2 + n0 / 2 - 1, aP1 = \*arr4, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }    //Запись первой коробки  for (aP = \*arr, end = \*arr + n0 / 2 - 1, aP1 = \*arr3, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись второй коробки  for (aP = \*arr + n0 / 2, end = \*arr + n0 - 1, aP1 = \*arr4, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись третьей коробки  for (aP = \*arr + n0 \* n0 / 2, end = \*arr + n0 \* n0 / 2 + n0 / 2 - 1, aP1 = \*arr1, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись четвертой коробки  for (aP = \*arr + n0 \* n0 / 2 + n0 / 2, end = \*arr + n0 \* n0 / 2 + n0 / 2 + n0 / 2 - 1, aP1 = \*arr2, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }    for (int l = 0; l < n0/2; l++)  {  for (int r = 0; r < n0/2; r++)  {  cout << arr1[l][r] << " ";  if (r == (n0/2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0/2; l++)  {  for (int r = 0; r < n0/2; r++)  {  cout << arr2[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0 / 2; l++)  {  for (int r = 0; r < n0 / 2; r++)  {  cout << arr3[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0 / 2; l++)  {  for (int r = 0; r < n0 / 2; r++)  {  cout << arr4[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  cout << "Вид массива после перестановки" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }    } |
|  |
| #include <iostream>  #include <iomanip>  #include <iostream>  #include <Windows.h>  #include <stdlib.h>  #include <time.h>  #include <chrono>  using namespace std;  int GetRand(int min, int max)  {  int num = min + rand() % (max - min + 1);  return num;  }  int main()  {  srand(time(NULL));  setlocale(LC\_ALL, "ru");  const int M = 6;  const int N = 3;  int arr[M][M];  int arr1[N][N];  int arr2[N][N];  int arr3[N][N];  int arr4[N][N];  int width = 3;  int k;  int\* first = nullptr;  int\* ptr = nullptr;  int\* ptr1 = nullptr;  int\* aP = nullptr;  int\* aP1 = nullptr;  int n0 = sizeof(arr) / sizeof(\*arr);  int m0 = sizeof(\*arr) / sizeof(\*\*arr);  int\* end = \*arr + m0 \* n0 - 1;  int\* end1 = nullptr;  int iter;  int n1, m1;  for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = GetRand(1, 36);  }    cout << "Исходный вид массива" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  //Сборка первой коробки  for (aP = \*arr, end = \*arr + n0 / 2 - 1, aP1 = \*arr1, k = 1; k <= n0/2; aP = aP + n0, end = end+n0, aP1=aP1+(n0/2), k++)  {    for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка второй коробки  for (aP = \*arr+n0/2, end = \*arr + n0 - 1, aP1 = \*arr2, k = 1; k <= n0 /2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка третьей коробки  for (aP = \*arr + n0\*n0/2, end = \*arr +n0\*n0/2+n0/2-1, aP1 = \*arr3, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }  //Сборка четвертой коробки  for (aP = \*arr + n0 \* n0 / 2+n0/2, end = \*arr + n0 \* n0 / 2 + n0 / 2 + n0 / 2 - 1, aP1 = \*arr4, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr1 = \*ptr;  }  }    //Запись первой коробки  for (aP = \*arr, end = \*arr + n0 / 2 - 1, aP1 = \*arr2, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись второй коробки  for (aP = \*arr + n0 / 2, end = \*arr + n0 - 1, aP1 = \*arr1, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись третьей коробки  for (aP = \*arr + n0 \* n0 / 2, end = \*arr + n0 \* n0 / 2 + n0 / 2 - 1, aP1 = \*arr4, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }  //Запись четвертой коробки  for (aP = \*arr + n0 \* n0 / 2 + n0 / 2, end = \*arr + n0 \* n0 / 2 + n0 / 2 + n0 / 2 - 1, aP1 = \*arr3, k = 1; k <= n0 / 2; aP = aP + n0, end = end + n0, aP1 = aP1 + (n0 / 2), k++)  {  for (ptr = aP, ptr1 = aP1; ptr <= end; ptr++, ptr1++)  {  \*ptr = \*ptr1;  }  }    for (int l = 0; l < n0/2; l++)  {  for (int r = 0; r < n0/2; r++)  {  cout << arr1[l][r] << " ";  if (r == (n0/2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0/2; l++)  {  for (int r = 0; r < n0/2; r++)  {  cout << arr2[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0 / 2; l++)  {  for (int r = 0; r < n0 / 2; r++)  {  cout << arr3[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  for (int l = 0; l < n0 / 2; l++)  {  for (int r = 0; r < n0 / 2; r++)  {  cout << arr4[l][r] << " ";  if (r == (n0 / 2 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";  cout << "Вид массива после перестановки" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }    } |
| *3)* Выполнить сортировку массива, используя арифметику указателей. Алгоритм сортировки массива – shaker sort |
| #include <iostream>  #include <iomanip>  #include <iostream>  #include <Windows.h>  #include <stdlib.h>  #include <time.h>  #include <chrono>  using namespace std;  int GetRand(int min, int max)  {  int num = min + rand() % (max - min + 1);  return num;  }  int main()  {  srand(time(NULL));  setlocale(LC\_ALL, "ru");  const int M = 6;  const int N = 3;  int arr[M][M];  int arr1[N][N];  int arr2[N][N];  int arr3[N][N];  int arr4[N][N];  int width = 3;  int k;  int\* first = nullptr;  int\* ptr = nullptr;  int\* ptr1 = nullptr;  int\* aP = nullptr;  int\* aP1 = nullptr;  int n0 = sizeof(arr) / sizeof(\*arr);  int m0 = sizeof(\*arr) / sizeof(\*\*arr);  int\* end = \*arr + m0 \* n0 - 1;  int\* end1 = nullptr;  int iter;  int n1, m1;  bool swapped = true;  int\* start = \*arr;  for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = GetRand(1, 36);  }    cout << "Исходный вид массива" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n" << "\n" << "\n";      while (swapped) {  swapped = false;  // Проход слева направо  for (int \*i = \*arr; i< end; ++i)  {  if (\*i > \*(i+1))  {  swap(\*i, \*(i+1));  swapped = true;  }  }  if (!swapped) {  break;  }  swapped = false;  --end;  // Проход справа налево  for (int\*i = end ; i >= start; --i)  {  if (\*i > \*(i + 1))  {  swap(\*i, \*(i + 1));  swapped = true;  }  }  ++start;  }    cout << "\n" << "\n" << "\n" << "\n";  cout << "Вид массива после сортировки" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }    } |
| *4)* Выполнить уменьшение/умножение/деление элементов матрицы на введенное пользователем число, используя только арифметику указателей |
| #include <iostream>  #include <iomanip>  #include <iostream>  #include <Windows.h>  #include <stdlib.h>  #include <time.h>  #include <chrono>  using namespace std;  int GetRand(int min, int max)  {  int num = min + rand() % (max - min + 1);  return num;  }  int main()  {  srand(time(NULL));  setlocale(LC\_ALL, "ru");  const int M = 6;  const int N = 6;  int arr[M][M];  int k;  int\* first = nullptr;  int\* ptr = nullptr;  int\* aP = nullptr;  int n0 = sizeof(arr) / sizeof(\*arr);  int m0 = sizeof(\*arr) / sizeof(\*\*arr);  int\* end = \*arr + m0 \* n0 - 1;  int iter;  int n1, m1;    for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = GetRand(1, 36);  }    cout << "Исходный вид массива" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n";  cout << "Введите произвольное целое число";  cin >> k;      for (ptr =\*arr; ptr <= end; ptr++)  {  \*ptr = \*ptr+k;  }    cout << "\n" << "\n";  cout << "Вид массива после увеличения на заданное число" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n";  for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = \*ptr -2\*k;  }  cout << "Вид массива после уменьшения на заданное число" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n";  for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = (\*ptr + k)\*k;  }  cout << "Вид массива после умножения на заданное число" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }  cout << "\n" << "\n";  for (ptr = \*arr; ptr <= end; ptr++)  {  \*ptr = \*ptr/k/k;  }  cout << "Вид массива после деления на заданное число" << endl;  for (int l = 0; l < n0; l++)  {  for (int r = 0; r < m0; r++)  {  cout << arr[l][r] << " ";  if (r == (n0 - 1))  cout << endl;  }  }    } |