**Лабораторная работа № 14. Многомерные массивы**

|  |  |  |
| --- | --- | --- |
| Задание | | Решение |
| 1. Выполнить программу, записанную в правой части. Внести изменения в программу с тем, чтобы инициализировался другой массив. Осуществить вывод этого массива в виде матрицы. | | **#include <iostream>**  **void main()**  **{ const int ci = 3, const int cj = 2;**  **int a[ci][cj] = { { 1, 2 }, { 3, 4 }, { 5, 6 } };**  **for (int i = 0; i < ci; i++)**  **for (int j = 0; j < cj; j++)**  **std::cout << "\n a[" << i << "," << j << "] =" << a[i][j];**  **}** |
| 2. Выполнить программы, записанные в правой части для одной и той же задачи, условие которой надо определить. Одна из программ использует указатели.*.* | | **Условие: Написать программу которая считает сумму элементов заданной матрицы** |
| 3. Изучить способы работы с двумерными массивами, выполнив программу в правой части и записав ее условие.  Внести изменения в программу с тем, чтобы определялись минимальные элементы каждой строки. Для доступа к элементам матрицы использовать указатели. | | **#include <iostream>**  **void main()**  **{ setlocale(LC\_CTYPE, "Russian");**  **using namespace std;**  **const int n = 2, m = 4;**  **int B[n][m];**  **int i, j, row = 0, column = 0;**  **cout << "Введите элементы массива" << endl;**  **for (i = 0; i < n; i++)**  **for (j = 0; j < m; j++)**  **cin >> B[i][j];**  **int min = B[0][0];**  **for (int i = 0; i < n; i++)**  **for (int j = 0; j < m; j++)**  **if (min > B[i][j])**  **{ min = B[i][j];**  **column = i;**  **row = j;**  **}**  **cout << " Исходный массив:" << endl;**  **for (int i = 0; i < n; i++)**  **{ cout << "\n";**  **for (int j = 0; j < m; j++)**  **cout << "B[" << i << "," << j << "] =" << B[i][j] << "\t";**  **}**  **cout << endl;**  **cout << "Минимальный элемент B[" << column << "," << row << "] ="<<min<< endl;**  **}**    #include <iostream>  void main()  {  setlocale(LC\_CTYPE, "Russian");  using namespace std;  const int n = 2, m = 4;  int B[n][m];  int i, j, row[n] = { 0,0 }, column[n] = { 1,0 };  cout << "Введите элементы массива" << endl;  for (i = 0; i < n; i++)  for (j = 0; j < m; j++)  cin >> B[i][j];  int min[n] = { B[0][0],B[1][0] };  for (int i = 0; i < n; i++)  for (int j = 0; j < m; j++)  if (min[i] > B[i][j])  {  min[i] = B[i][j];  column[i] = i;  row[i] = j;  }  cout << " Исходный массив:" << endl;  for (int i = 0; i < n; i++)  {  cout << "\n";  for (int j = 0; j < m; j++)  cout << "B[" << i << "," << j << "] =" << B[i][j] << "\t";  }  cout << endl;  cout << "Минимальный элемент B[" << column[0] << "," << row[0] << "] =" << min[0] << endl;  cout << "Минимальный элемент B[" << column[1] << "," << row[1] << "] =" << min[1] << endl;  } |
| 4. Выполнить задания из таблицы ниже, используя ***индексы*** для доступа к элементам массивов в первой программе и ***указатели*** − во второй программе. | | |
| №Варианта  10 | | |
| Задание | Код/Результат | |
| 1. Дана матрица **В(n, m).** Вычислить произведение чётных положительных элементов матрицы. | #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "rus");  int n, m;  cin >> n >> m;  int a[10][10];  int sum = 1;  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cin >> a[i][j];  if (a[i][j] > 0 && a[i][j] % 2 == 0) {  sum \*= a[i][j];  }  }  }  cout << "Произведение элементов: " << sum;  return 0;  } | |
| 2. Проверить, есть ли в матрице хотя бы одна строка, содержащая элемент, равный нулю, и найти ее номер. Уменьшить все элементы матрицы на значение первого элемента найденной строки.). | #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "rus");  int n, m;  cin >> n >> m;  int a[10][10];  int i1 = -1;  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cin >> \*(\*(a + i) + j);  if (\*(\*(a+i)+j) == 0) {  i1 = i;  }  }  }  if (i1 == -1) {  cout << "Нет строки с 0";  return 0;  }  cout << "Строка номер: " << i1 + 1 << '\n';  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  \*(\*(a + i) + j) -= \*\*(a + i1);  }  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < m; j++) {  cout << \*(\*(a + i) + j) << ' ';  }  cout << endl;  }  return 0;  } | |

|  |  |
| --- | --- |
| Допы | |
| Задание | Код/результат |
| 1. Дана квадратная матрица порядка **2n**, элементы которой формируются случайным образом и находятся в пределах от −10 до 10. Получить новую матрицу, переставляя ее блоки размера **n×n** в соответствии со схемой. | srand(time(0));  int n;  cin >> n;  int a[20][20];  cout << "Исходный массив:\n";  for (int i = 0; i < 2 \* n; i++) {  for (int j = 0; j < 2 \* n; j++) {  a[i][j] = (rand() % 21) - 10;  cout << a[i][j] << ' ';  }  cout << '\n';  }  cout << "Изменённый массив:\n";  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  int temp = a[i][j];  a[i][j] = a[i + n][j + n];  a[i + n][j + n] = temp;  }  }  for (int i = n; i < 2 \* n; i++) {  for (int j = 0; j < n; j++) {  int temp = a[i][j];  a[i][j] = a[i - n][j + n];  a[i - n][j + n] = temp;  }  }  for (int i = 0; i < 2 \* n; i++) {  for (int j = 0; j < 2 \* n; j++) {  cout << a[i][j] << ' ';  }  cout << '\n';  }  return 0; |
| 2. Латинским квадратом порядка **n** называется квадратная таблица размером **nхn**, каждая строка и каждый столбец которой содержат все числа от 1 до **n**. Для заданного **n** в матрице **L(n, n**) построить латинский квадрат порядка **n**. | int n;  cin >> n;  int a[20][20];  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  a[i][j] = (j + i) % n + 1;  }  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << a[i][j] << ' ';  }  cout << '\n';  } |
| 3. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине − в позиции (2, 2), следующий − в позиции (3, 3) и т. д., заполнив таким образом всю главную диагональ. | int n;  cin >> n;  float a[10][10];  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cin >> a[i][j];  }  }  for (int k = 0; k < n; k++) {  int maxi = k, maxj = k;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if(i==j && i<=k) continue;  if (a[maxi][maxj] < a[i][j]) {  maxi = i;  maxj = j;  }  }  }  int temp = a[k][k];  a[k][k] = a[maxi][maxj];  a[maxi][maxj] = temp;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << a[i][j] << ' ';  }  cout << '\n';  } |