Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 14

По дисциплине «Основы алгоритмизации и программирования»

На тему «Многомерные массивы»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условие задачи** |
| --- | --- |
| **9** | 1. Дана матрица **A(n, n).** Переписать элементы её главной диагонали в одномерный массив **Y(N)** и разделить их на максимальный элемент главной диагонали.  2. Найти в матрице первый столбец, все элементы которого положительны. Знаки элементов предыдущего столбца изменить на противоположные. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  srand((unsigned)time(NULL));  const int size = 100;  float array[size][size],y[size],n,max\_el=0;//  cout << "Введите размер матрицы:";  cin >> n;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  array[i][j] = (rand() % 100)+1;//за шаг цикла j мы проходим один элемент массива по горизонтали,а за шаг цикла i перемещаемся по вертикали нашего квадрата  max\_el = max(max\_el, array[i][i]);//находим максимальный элемент главной диагонали  }  }  for (int i = 0; i < n; i++) {  for (int j=0; j < n; j++) {  cout<<array[i][j]<<" ";//вывдим получивщуюся квадартную матрицу  }  cout << endl;  }  cout << endl << "Максимальный элемент главной диагонали:" << max\_el << endl;  for (int i = 0; i < n; i++) {  y[i] = array[i][i] / max\_el;//переписываем элементы главной диагонали,деленные на макс. элемент  }  for (int i = 0; i < n; i++) {  cout<< y[i]<<" ";//вывдим получившийся массив  }  return 0;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  srand((unsigned)time(NULL));  const int size = 100;  int array[size][size],n;  int ind\_pos ;  cout << "Введите размер матрицы:";  cin >> n;  ind\_pos = n + 1;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  array[i][j] = (rand() % 100)-15;//за шаг цикла j мы проходим один элемент массива по горизонтали, а за шаг цикла i перемещаемся по вертикали нашего квадрата  }  }  for (int i = 0; i < n; i++) {  for (int j=0; j < n; j++) {  cout<<array[i][j]<<" ";//выводим получившуюся квадратную матрицу  }  cout << endl;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  if (array[j][i] < 1) { break; }//если встретили неположительное число, то переходим к следующему столбцу  if(j==n-1){//если до шли до последнего элемента, то запоминаем номер столбца и заканчиваем анализ  ind\_pos = i;  i = n;  }  }  }  if (ind\_pos > n) { cout << "Положительных столбцов нет"; return 0; }//если мы не встретили положительных столбцов, то заканчиваем программу  if(ind\_pos!=0)//если положительный столбец не первый, то меняем предшествующий  {  for (int j = 0; j < n; j++) {  array[j][ind\_pos - 1] \*= -1;  }  }  cout << endl;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << array[i][j] << " ";//выводим получившуюся квадратную матрицу  }  cout << endl;  }  return 0;  } |
| **Результат программы** |
|  |

Дополнительные задания

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.

|  |
| --- |
| **Код программы** |
| #include <ctime>  #include <iostream>  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  srand((unsigned)time(NULL));  const int size =100;  int n;  int array[size][size];  cout << "Введите размер матрицы:";  cin >> n;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  array[i][j] = (rand() % 20) - 10;//за шаг цикла j мы проходим один элемент массива по горизонтали, а за шаг цикла i перемещаемся по вертикали нашего квадрата  }  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << array[i][j] << " ";//выводим получившуюся квадратную матрицу  }  cout << endl;  }  cout << endl << "Обновленная матрица" << endl;  for (int i = n - 1; i > -1; i--) {  for (int j = n - 1; j > -1; j--) {  cout << array[i][j] << " ";//выводим квадратную матрицу в обратном порядке  }  cout << endl;  }  return 0;  } |
| **Результат программы** |
|  |

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**.

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  srand((unsigned)time(NULL));  const int size = 100;  int array[size][size],n;  cout << "Введите размер матрицы:";  cin >> n;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  array[i][j] = (i+j)%n+1;//добавляем единицу, что генерировались числа от 1,а не 0 . используя индексы мы находим значение для каждого элемента так, чтобы он не повторялся в своём столбце и строке  }  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << array[i][j] << " ";//выводим получившуюся квадратную матрицу  }  cout << endl;  }  return 0;  } |
| **Результат программы** |
|  |

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.

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  int main() {  system("color 70");  setlocale(LC\_CTYPE, "Russian");  srand((unsigned)time(NULL));  const int size = 100;  int array[size][size],n,max\_el=0,mindif=0,max\_el2=0;  cout << "Введите размер матрицы:";  cin >> n;  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  array[i][j] = rand() % 100;//добавляем единицу, что генерировались числа от 1,а не 0 . используя индексы мы находим значение для каждого элемента так, чтобы он не повторялся в своём столбце и строке  max\_el = max(max\_el, array[i][j]);//ищем максимальный элемент во всей матрице  }  }    for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << array[i][j] << " ";//выводим получившуюся квадратную матрицу  }  cout << endl;  }  cout << endl;  array[0][0] = max\_el;//зная максимальный элемент всей матрицы, запишем его в первую ячейку главной диагонали  for(int k=1;k<n;k++){  for (int i = 0; i < n; i++) {//проходим по всей матрице столько раз, сколько ячеек в главной диагонали нам надо заполнить  for (int j = 0; j < n; j++) {  if (max\_el - array[i][j] < max\_el - max\_el2 and max\_el-array[i][j]>0) { max\_el2 = array[i][j]; }//находим следующий по величине элемент  }  }  array[k][k] = max\_el2;//перестанавливаем его в ячейку главной диагонали  max\_el = max\_el2;//принимаем его за максимум, чтобы найти следующего за ним по значению  max\_el2 = 0;  }  for (int i = 0; i < n; i++) {  for (int j = 0; j < n; j++) {  cout << array[i][j] << " ";//выводим новую квадратную матрицу  }  cout << endl;  }  return 0;  } |
| **Результат программы** |
|  |