|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
|  | | |
| Расчетно-графическое задание | | |
| по практикуму на ЭВМ: методы программирования | | |
|  | | |
|  | | |
|  | | |
|  | Факультет: | ФПМИ |
| Группа: | ПМИ-12 |
|  |  |
| Студент: | Мироненко А. М. |
|  |
|  |
| Преподаватель: | Филиппова Е. В. |
|  |  |
|  | | |
| Новосибирск | | |
| 2021 | | |

**1. Условие задачи (вариант 59)**

По заданной квадратной матрице размером 10\*10 построить вектор длиной 19, элементы которого – максимумы элементов диагоналей, параллельных главной диагонали.

**2. Анализ задачи**

**Дано:** Размерность матрицы – 10\*10. Некоторая действительная матрица

**Результат:** Программа выведет вектор длиной 19, содержащий максимумы элементов диагоналей, параллельных главной диагонали.  
**Анализ:** для того, чтобы решить данную задачу, необходимо поочередно найти максимальные элементы всех диагоналей, параллельных главной, после чего записать их в отдельный вектор-массив. Также следует продумать какой-либо способ получения массива с числами, т. к. вводить 100 чисел с клавиатуры не очень оптимально, можно использовать функцию srand(time(NULL)).

Для определения элементов, лежащих на одной диагонали воспользуемся следующими формулами.  
До главной диагонали включительно:

Зафиксируем номер столбца (i), тогда элементы, лежащие на одной диагонали, будут выражаться как mass[i][i - size + l], где i – номер столбца, size – размерность массива (10), l – номер рассматриваемой диагонали (счет с левого нижнего угла).

Выше главной диагонали:

Зафиксируем номер столбца (i), тогда элементы, лежащие на одной диагонали, будут выражаться как [i][i + l - size], где i – номер столбца, size – размерность массива (10), l – номер рассматриваемой диагонали.

**Способ решения задачи:**

Проанализировав условие задачи, я решила прописать два варианта получения массива, состоящего из целых чисел: с помощью ввода с клавиатуры (вложенные циклы for) и с автоматическим заполнением случайными числами (подключение библиотек **<cstdlib>** и **<ctime>,** использование функции **srand(time(NULL)).**

После чего были написаны вложенные циклы **for** для вывода, чтобы наглядно видеть полученный массив.

*Я написала универсальный вариант программы, ее можно использовать не только для матрицы 10\*10, но и для любой другой, если присвоить соответствующее значение переменной* ***const int size****.*

Далее я написала цикл **for** для постепенного «прохождения» по всем диагоналям. Данный цикл делился на две часть условными операторами **if** и **else**. Поиск максимумов диагоналей начинается с нижнего левого угла до главной диагонали (первая часть с **if**), далее идет вторая часть – от главной диагонали до правого верхнего угла (часть с **else**). В каждой части был прописан цикл **for** со вложенным условием **if** для поиска максимального элемента на диагонали. В каждой из частей максимальный элемент на диагонали записывается (начиная с левого нижнего угла, заканчивая правым верхним) в новый массив **vector** длиной 19.   
После чего данный вектор-массив выводится на экран посредством еще одного цикла **for.**

**Использование/отсутствие подпрограмм:**  
В ходе решения задачи подпрограммы не были задействованы. Была использована лишь одна функция **srand(time(NULL)).**

**3. Структуры данных, используемые для представления исходных данных и результатов задачи**  
Входные данные.

Внешние:

Нет.

Внутренние (все целого типа):

Переменные целочисленного типа:

Константы:

const int size = 10

*Использование константного значения обеспечивает универсальность программы, достаточно поменять размерность в одном месте, а не во всем коде.*

Массивы:

int mass[size][size] – двумерный

*Данная форма представления была выбрана согласно условию задачи (задана матрица).*

Переменные:

int maxl;

int n;

Выходные данные.

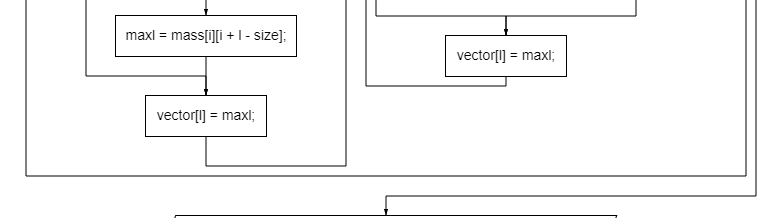
Внешние данные (массив):

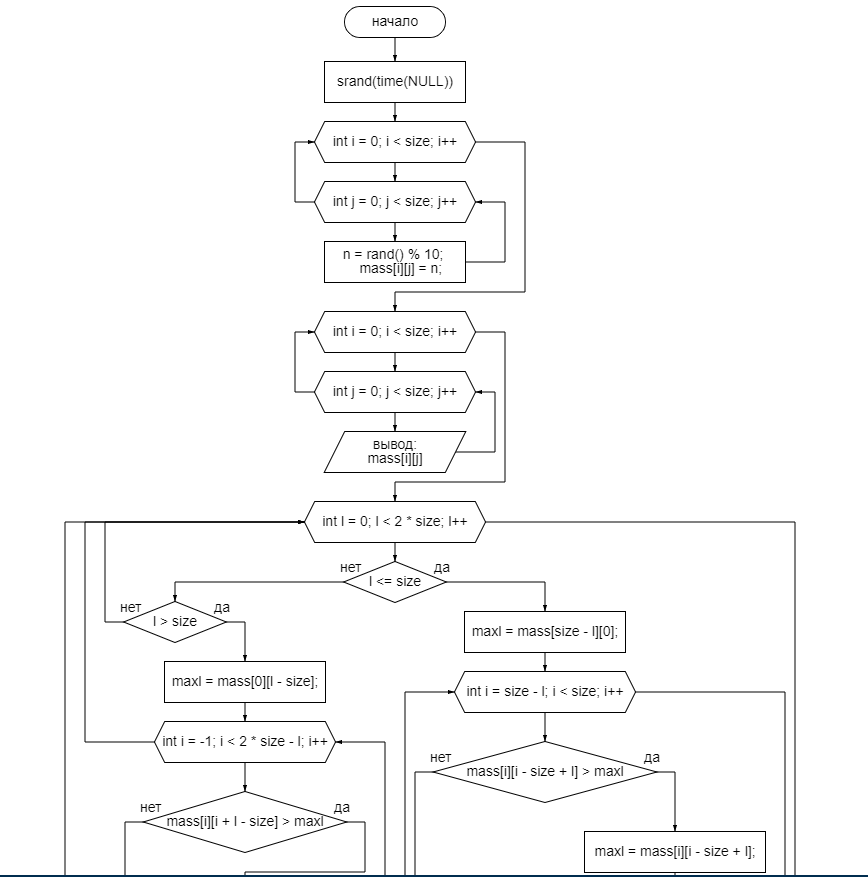
int vector[2 \* size - 1] - одномерный

*Была выбрана именно такая форма представления выходных данных, потому что в условии задачи сказано, что необходимо построить вектор (вектор – частный случай массива, одномерный массив).*

**4. Алгоритм решения задачи**

1. Получим матрицу 10\*10, заполненную произвольными числами.
2. Выведем эту матрицу для наглядности и проверки правильности решения.
3. «Закрепим» одну диагональ.
4. Если порядковый номер диагонали меньше, чем ее размерность (10), то будем считать, что мы движемся от левого нижнего угла до главной диагонали включительно. Тогда присвоим временной переменной для определения максимума значение левой нижней угловой переменной, поочередно сравнивая временную переменную со значениями, которые принадлежат данной диагонали, определим максимум и «запишем» его в вектор-массив.
5. Если порядковый номер диагонали больше, чем размерность массива (10), будем считать, что мы движемся от главной диагонали до правого верхнего угла. Тогда присвоим временной переменной для определения максимума значение из данного массива. После чего, поочередно сравнивая элементы данной диагонали с временной переменной, определим максимум и «запишем» его в вектор-массив.
6. Продолжая этот процесс, мы переберем все диагонали и, соответственно, определим на каждой максимум и «запишем» его в вектор-массив.
7. Далее выведем полученный вектор (длиной 19).

****

****

****

**4. Структура программы.**

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#include <string.h>

#include <ctime>

int main()

{

setlocale(LC\_CTYPE, "Russian");

const int size = 10; // размерность

int mass[size][size]; // массив 10\*10

int vector[2 \* size - 1]; // вектор максимумов

int maxl; // переменная для поиска максимумов

int n;

srand(time(NULL)); //заполнение массива случайными числами

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

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

n = rand() % 10;

mass[i][j] = n;

}

}

//printf("Введите массив\n");

//for (int i = 0; i < size; i++) { //заполняем массив вручную

//for (int j = 0; j < size; j++) {

//scanf\_s("%d", &mass[i][j]);

//}

//}

for (int i = 0; i < size; i++) { //вывод значений

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

printf("%d\t", mass[i][j]);

}

printf("\n");

}

printf("\n");

for (int l = 0; l < 2 \* size; l++) { // кол-во диагоналей

if (l <= size) { // идем из левого нижнего угла до главной диагонали

maxl = mass[size - l][0];

for (int i = size - l; i < size; i++) {

if (mass[i][i - size + l] > maxl) {

maxl = mass[i][i - size + l];

}

vector[l] = maxl;

}

}

else {

if (l > size) { // идем вправо до правого верхнего угла

maxl = mass[0][l - size];

for (int i = -1; i < 2 \* size - l; i++) {

if (mass[i][i + l - size] > maxl) {

maxl = mass[i][i + l - size];

}

vector[l] = maxl;

}

}

}

}

printf("Максимальные элементы диагоналей, параллельных главной:\n");

for (int i = 1; i < 2 \* size; i++) {

printf("%d\t", vector[i]);

}

return 0;

}

**5. Тесты.**

|  |  |  |
| --- | --- | --- |
| **№** | **Матрица** | **Результат** |
| **1** | 0 6 9 2 0 5 4 9 9 8  1 3 5 7 9 3 6 3 8 5  6 3 4 0 9 8 1 8 8 9  7 3 3 7 6 6 4 5 2 6  1 6 0 6 7 1 5 7 5 3  2 3 6 8 5 1 2 9 5 6  3 8 0 0 0 1 2 0 5 9  0 0 6 6 4 5 3 4 6 3  9 2 9 1 1 3 5 3 8 7  0 5 3 1 2 4 4 5 5 2 | 0 9 3 9 8 6 7 8 6 8 7 9 9 6 8 89 9 8 |
| **2** | 13 20 98 43 98 19 8 35 47 46  1 25 86 25 86 34 31 38 96 52  7 52 77 21 38 56 22 24 28 40  44 7 66 48 84 54 35 82 75 55  85 9 1 15 98 82 60 12 6 75  77 84 35 27 39 64 85 37 3 80  63 60 14 58 56 60 48 37 66 74  10 52 85 26 63 59 29 0 48 93  1 71 56 55 63 70 64 92 8 27  20 89 69 54 14 23 70 82 40 67 | 20 89 71 63 85 85 70 82 92 98 86 98 86 98 75 55 96 52 46 |
| **3** | 25 70 89 63 17 12 10 70 56 38  55 88 21 61 85 29 97 62 21 27  51 5 1 17 59 81 22 68 11 81  98 39 47 22 59 72 87 25 99 56  14 22 81 63 65 81 72 38 53 68  15 92 29 93 37 12 16 66 66 83  93 74 53 5 55 69 99 18 31 50  15 77 44 56 40 45 39 92 1 30  35 2 51 63 10 86 52 97 8 83  45 74 32 84 25 7 92 83 29 99 | 45 74 32 93 74 92 98 93 97 99 83 89 87 83 99 62 81 56 38 |

**6. Результат работы.**

На всех тестах программа выдала ожидаемый результат, следовательно, задача решена верно.  
Стоит отметить, что программа написана для целых чисел, но ее можно легко использовать и для вещественных, если изменить тип переменных, также можно настроить любую размерность квадратной матрицы. Таким образом, данная программа универсальна в использовании.