**Язык СИ**

**Двумерные массивы**

**Задание :**

**1)**Соседями элемента Аij в матрице назовем элементы Аkl с i-1  к  i+1, j-1  l  j+1, (k, l) (i,j). Операция сглаживания матрицы дает новую матрицу того же разме­ра, каждый элемент которой получается как среднее арифметическое имеющих­ся соседей соответствующего элемента исходной матрицы. Построить результат сглаживания заданной вещественной матрицы размером 10 на 10. В сглаженной матрице найти сумму модулей элементов, расположенных ниже главной диагонали.

Сделать с вводом с файла (Код)

**Решение :**

#include <stdio.h>

#include <math.h>

int main(int argc, char\* argv[])

{

/\* Объявляем и инициализируем матрицу 10х10 \*/

float matrix[10][10] = {

{ 2, 7.3, 2, 7, 1, 4, 0, 3, 1.5, 9.8 },

{ 1.7, 0, 7, 6, 0, -8, 3, 5, -5.6, 8.8 },

{ 2.9, 7, 1, 8, 1, 4, 9, 3, 4.3, -2.7 },

{ 9, -2.5, 8.2, 5, 2, 0, 0, 6, 7.7, 8.1 },

{ 7, 1, -1, 3, 9, 3.2, -9.8, 1, 1.6, 6.3 },

{ 8.05, -2, 4.92, 9, 1, -6, 4, 9, 5.5, -2.9 },

{ 0, 3, 9, 0, 9, 4, -8, 8, 6.4, 9.5 },

{ 1, -8, 3, 2, -8, 2, 8, 0, -2.2, 3.4 }

{ 4.37, 7, 5, -8.4, 1.9, 4, 9, 3, 4.6, -7.7 },

{ 0.5, 3.8, -4.9, 0, 9, 4, 8, 8, 6.4, 9.5 },

};

int i, j; /\* Счетчики \*/

/\* Переменные и массив в который будет записана сглаженная вещественная матрица,\*/

float iminus1, iplus1, jminus1, jplus1, newmatrix[10][10], absfSumm=0;

/\*----------------------------------------------\*/

/\* П Е Р В А Я Ч А С Т Ь З А Д А Н И Я \*/

/\*----------------------------------------------\*/

printf("\n"); /\* Перевод курсора на новую строку \*/

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

{

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

{

/\* Переменной iminus1 присваивается значение А i-1,j или значение 0 если i-я строка первая (в массиве она нулевая)

Переменной iplus1 присваивается значение А i+1,j или значение 0 если

i-я строка последняя (в массиве она девятая)

Переменной jminus1 присваивается значение А i,j-1 или значение 0 если j-й столбец первый (в массиве он нулевой)

Переменной jplus1 присваивается значение А i,j+1 или значение 0 если j-й столбец последний(в массиве он девятый)\*/

if (i==0)

{

iminus1=0;

}

else

{

iminus1= matrix[i-1][j];

}

if (i==9)

{

iplus1=0;

}

else

{

iplus1= matrix[i+1][j];

}

if (j==0)

{

jminus1=0;

}

else

{

jminus1= matrix[i][j-1];

}

if (j==9)

{

jplus1=0;

}

else

{

jplus1= matrix[i][j+1];

}

newmatrix[i][j]= (iminus1+ iplus1+ jminus1+ jplus1)/4;

printf("%f ", newmatrix[i][j]); /\* Вывод на экран элемента сглаженной вещественной матрицы, после которого будет стоять пробел \*/

}

printf("\n"); /\* Перевод курсора на новую строку \*/

}

/\* В Т О Р А Я Ч А С Т Ь З А Д А Н И Я \*/

/\*----------------------------------------------\*/

printf("\n\n");

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

/\* В цикле перебираем все строки \*/

{

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

/\* В цикле перебираем элементы i-й строки изменяя номер столбца \*/

{

if ( i > j)

/\* У элементов, расположенных ниже главной диагонали номер строки всегда больше номера столбца. Поэтому выше проводим проверку, если i > j к переменной absfSumm добавляем модуль элемента матрицы, если нет переход к началу цикла где j увеличивается на 1 \*/

absfSumm += abs (matrix[i][j]);

/\* Функция abs() возвращает модуль числа \*/

}

}

printf("Сумма элементов, расположенных ниже главной диагонали равняется %f \n", absfSumm);

/\* выводим на экран сумму модулей элементов, расположенных ниже главной диагонали \*/

return 0;

}