#include <stdio.h>

void aufgabe1(void)

{

int data1[] = {3, 2, 5};

int data2[] = {1, 4, 4, 8, 13};

int data3[] = {9, 1, 0, 2};

int data4[] = {0, 2, 6, 3, -1, -8};

const int COL1 = 3;

const int COL2 = 5;

const int COL3 = 4;

const int COL4 = 6;

const int ROW = 4;

int col[] = {COL1, COL2, COL3, COL4};

int \*arr[] = {data1, data2, data3, data4};

int i = 0, k = 0;

int sumreihe = 0, sum = 0;

int maxarr = 0, minarr = 0, max = 0;

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

{

sumreihe = 0; //Alles auf Null, damit bei nächster reihe

k=0; //wieder neu berechnet werden kann

max = arr[i][k];

for(k = 0; k < col[i]; k++)

{

sumreihe += arr[i][k];

sum += arr[i][k]; //Summe aller Feldelemente

if(max < arr[i][k]) //MAX, Suche

max = arr[i][k];

if(maxarr < arr[i][k])//MAX einer Reihe

maxarr = arr[i][k];

if(minarr > arr[i][k])//MIN einer Reihe

minarr = arr[i][k];

}

printf("Zeile %i -> Größtes Feldelement: %i, Summe der Feldelemente: %i \n", i, max, sumreihe);

}

printf("Summe der Feldelemente: %i\n", sum);

printf("Größtes Feldelement: %i\n", maxarr);

printf("Kleinstes Feldelement: %i\n", minarr);

}

void aufgabe2(void) //FELD UMDREHEN

{

int data1[] = {3, 2, 5};

int data2[] = {1, 4, 4, 8, 13};

int data3[] = {9, 1, 0, 2};

int data4[] = {0, 2, 6, 3, -1, -8};

const int COL1 = 3;

const int COL2 = 5;

const int COL3 = 4;

const int COL4 = 6;

const int ROW = 4;

int col[] = {COL1, COL2, COL3, COL4};

int \*arr[] = {data1, data2, data3, data4};

int help[6]; //Hilffeld, hier wird kurz das neue Feld gespeichert

int i, j;

for(i = 0; i < ROW; i++) //Reihen durchgehen

{

for(j = 0; j < col[i]; j++) //Elemente eines Feldes durchgehen

{//Zwei Elemente vertauschen.

help[j] = arr[i][col[i] - j -1];

printf("%i ", help[j]);

}

printf("\n");

}

}

void aufgabe3(void) //Spaltensumme mit Hilfsfeld

{

int data1[] = {3, 2, 5};

int data2[] = {1, 4, 4, 8, 13};

int data3[] = {9, 1, 0, 2};

int data4[] = {0, 2, 6, 3, -1, -8};

const int COL1 = 3;

const int COL2 = 5;

const int COL3 = 4;

const int COL4 = 6;

const int ROW = 4;

int col[] = {COL1, COL2, COL3, COL4};

int \*arr[] = {data1, data2, data3, data4};

int help[] = {0, 0, 0, 0, 0, 0};

int i, j;

for(i = 0; i < ROW; i++)//Reihen Durchgehen

{

for(j = 0; j < col[i]; j++)

{//Elemente aus reihe nacheinander in verschiedene Plätze im Feld

help[j] += arr[i][j];

}

}

for(i = 0; i < 6; i++) //Ausgabe Hilfsfeld

{

printf("Summe Spalte %i: %i\n", i+1, help[i]);

}

}

void aufgabe3b(void) //Spaltensumme ohne Hilfsfeld

{

int data1[] = {3, 2, 5};

int data2[] = {1, 4, 4, 8, 13};

int data3[] = {9, 1, 0, 2};

int data4[] = {0, 2, 6, 3, -1, -8};

const int COL1 = 3;

const int COL2 = 5;

const int COL3 = 4;

const int COL4 = 6;

const int ROW = 4;

int col[] = {COL1, COL2, COL3, COL4};

int \*arr[] = {data1, data2, data3, data4};

int sum = 0;

int i, j;

//i ist die Reihe und j die Spalte der Felder

for(j = 0; j < 6; j++)//Aufzählen der Spalten

{

sum = 0;

for(i = 0; i < ROW; i++) //Aufzählen der Reihen

{

if(j < col[i])//Wenn Spalte größer ist als es Elemente im Feld gibt

sum += arr[i][j]; //dann Element nicht beachten!

}

printf("Summe Spalte %i: %i\n", j+1, sum);

}

}

void aufgabe4(void)

{

#define IROW 12

#define ICOL 12

int image [IROW][ICOL] = {{5,0,0,0,0,5,5,0,0,0,0,5},

{0,5,0,0,0,5,5,0,0,0,5,0},

{0,0,5,5,5,5,5,5,5,5,0,0},

{0,0,5,5,5,5,5,5,5,5,0,0},

{0,0,5,5,5,9,9,5,5,5,0,0},

{5,5,5,5,9,9,9,9,5,5,5,5},

{5,5,5,5,9,9,9,9,5,5,5,5},

{0,0,5,5,5,9,9,5,5,5,0,0},

{0,0,5,5,5,5,5,5,5,5,0,0},

{0,0,5,5,5,5,5,5,5,5,0,0},

{0,5,0,0,0,5,5,0,0,0,5,0},

{5,0,0,0,0,5,5,0,0,0,0,5}};

int r,c;

float mittelwert = 0, sum = 0;

printf("Originalbild:\n");

for(r = 0; r < IROW; r++)

{

for(c = 0; c < ICOL; c++)

{

printf("[%i] ", image[r][c]);

}

printf("\n");

}

printf("\nGeglaettetes Bild:\n\n");

for(r = 0; r < IROW; r++)

{

for(c = 0; c < ICOL; c++)

{

if(c!=0 && c!=11 && r!=0 && r!=11)

{

sum = image[r-1][c-1] + image[r-1][c ] + image[r-1][c+1] +

image[r ][c-1] + image[r ][c ] + image[r ][c+1] +

image[r+1][c-1] + image[r+1][c ] + image[r+1][c+1];

mittelwert = sum / 9;

}

if(r == 0)

{

sum = image[r ][c-1] + image[r ][c ] + image[r ][c+1] +

image[r+1][c-1] + image[r+1][c ] + image[r+1][c+1];

mittelwert = sum / 6;

}

if(c == 0)

{

sum = image[r-1][c ] + image[r-1][c+1] +

image[r ][c ] + image[r ][c+1] +

image[r+1][c ] + image[r+1][c+1];

mittelwert = sum / 6;

}

if(r == 11)

{

sum = image[r-1][c-1] + image[r-1][c ] + image[r-1][c+1] +

image[r ][c-1] + image[r ][c ] + image[r ][c+1];

mittelwert = sum / 6;

}

if(c == 11)

{

sum = image[r-1][c-1] + image[r-1][c ] +

image[r ][c-1] + image[r ][c ] +

image[r+1][c-1] + image[r+1][c ];

mittelwert = sum / 6;

}

if(r == 0 && c == 0)

{

sum = image[r ][c ] + image[r ][c+1] +

image[r+1][c ] + image[r+1][c+1];

mittelwert = sum / 4;

}

if(r == 0 && c == 11)

{

sum = image[r ][c-1] + image[r ][c ] +

image[r+1][c-1] + image[r+1][c ];

mittelwert = sum / 4;

}

if(r == 11 && c == 0)

{

sum = image[r-1][c ] + image[r-1][c+1] +

image[r ][c ] + image[r ][c+1];

mittelwert = sum / 4;

}

if(r == 11 && c == 11)

{

sum = image[r-1][c-1] + image[r-1][c ] +

image[r ][c-1] + image[r ][c ];

mittelwert = sum / 4;

}

mittelwert = round(mittelwert);

printf("[%.0f] ", mittelwert);

}

printf("\n");

}

}

int main(void)

{

aufgabe4();

return 0;

}