# Обработка на масиви. Работа с масиви и указатели. Обмен на данни (базови типове, масиви) между функции. Разработване на програми с масиви.

# Многомерни масиви

Многомерният масив представлява масив от масиви. В този си вид при двумерен масив всеки един елемент от едномерния масив съдържа в себе си още един масив. Представете си го като кутия, която съдържа в себе си други кутии, а на свой ред те могат да съдържат или други кутии или предмети. Няма ограничение за измерноста на масива – може да бъде двумерен, тримерен, четиримерен и т.н.

Основният прототип на деклариране на масив:

**<тип> име [измерение 1][измерение 2]…[ измерение N]**

Например декларация на тримерен масив би изглеждала така:

***int array[4][5][2]***

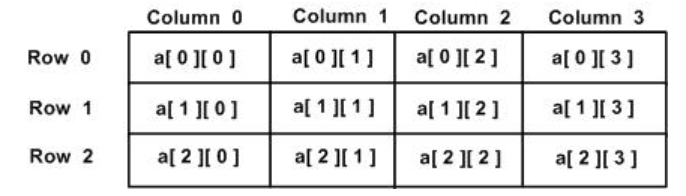
Това би създало масив с 4 реда всеки от тях с по 5 колони, а във всяка колона ще има по 2 елемента от тип цяло число.

# Двумерни масиви

Най – простата форма на многомерен масив е двумерният. Най – простото обяснение е, че това е списък с едномерни масиви, но често се оприличава с матрица. За да декларираме двумерен масив с X реда и Y колони трябва да напишем следния код:

**int array[x][y];**

Следната таблица показва размерността и използването на масив “a” с 3 реда и 3 колони



# 

# Инициализиране на масив

int a[3][4] = {

{0, 1, 2, 3} , /\* задаване на ред с индекс 0 \*/

{4, 5, 6, 7} , /\* задаване на ред с индекс 1 \*/

{8, 9, 10, 11} /\* задаване на ред с индекс 2 \*/

};

Може да се инициализира и по следния начин:

int a[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11};

# достъп по елементите на масив

за достъп до елементите на масив трябва да покажем в коя клетка искаме да влезем.

int val = a[2][3];

#include <stdio.h>

int main () {

/\* масив с 5 реда и 2 колони\*/

int a[5][2] = { {0,0}, {1,2}, {2,4}, {3,6},{4,8}};

int i, j;

/\* изкарване на всеки елемент \*/

for ( i = 0; i < 5; i++ ) {

for ( j = 0; j < 2; j++ ) {

printf("a[%d][%d] = %d\n", i,j, a[i][j] );1

}

}

return 0;

}

Различни начини за инициализация на масив:

int c[2][3] = {{1, 3, 0}, {-1, 5, 9}};

int c[][3] = {{1, 3, 0}, {-1, 5, 9}};

int c[2][3] = {1, 3, 0, -1, 5, 9};

тримерен масив:

int test[2][3][4] = {

{{3, 4, 2, 3}, {0, -3, 9, 11}, {23, 12, 23, 2}},

{{13, 4, 56, 3}, {5, 9, 3, 5}, {3, 1, 4, 9}}};

Примерни кодове:

// C program to store temperature of two cities of a week and display it.

#include <stdio.h>

const int CITY = 2;

const int WEEK = 7;

int main()

{

int temperature[CITY][WEEK];

// Using nested loop to store values in a 2d array

for (int i = 0; i < CITY; ++i)

{

for (int j = 0; j < WEEK; ++j)

{

printf("City %d, Day %d: ", i + 1, j + 1);

scanf("%d", &temperature[i][j]);

}

}

printf("\nDisplaying values: \n\n");

// Using nested loop to display vlues of a 2d array

for (int i = 0; i < CITY; ++i)

{

for (int j = 0; j < WEEK; ++j)

{

printf("City %d, Day %d = %d\n", i + 1, j + 1, temperature[i][j]);

}

}

return 0;

}

// C program to find the sum of two matrices of order 2\*2

#include <stdio.h>

int main()

{

float a[2][2], b[2][2], result[2][2];

// Taking input using nested for loop

printf("Enter elements of 1st matrix\n");

for (int i = 0; i < 2; ++i)

for (int j = 0; j < 2; ++j)

{

printf("Enter a%d%d: ", i + 1, j + 1);

scanf("%f", &a[i][j]);

}

// Taking input using nested for loop

printf("Enter elements of 2nd matrix\n");

for (int i = 0; i < 2; ++i)

for (int j = 0; j < 2; ++j)

{

printf("Enter b%d%d: ", i + 1, j + 1);

scanf("%f", &b[i][j]);

}

// adding corresponding elements of two arrays

for (int i = 0; i < 2; ++i)

for (int j = 0; j < 2; ++j)

{

result[i][j] = a[i][j] + b[i][j];

}

// Displaying the sum

printf("\nSum Of Matrix:");

for (int i = 0; i < 2; ++i)

for (int j = 0; j < 2; ++j)

{

printf("%.1f\t", result[i][j]);

if (j == 1)

printf("\n");

}

return 0;

}

// C Program to store and print 12 values entered by the user

#include <stdio.h>

int main()

{

int test[2][3][2];

printf("Enter 12 values: \n");

for (int i = 0; i < 2; ++i)

{

for (int j = 0; j < 3; ++j)

{

for (int k = 0; k < 2; ++k)

{

scanf("%d", &test[i][j][k]);

}

}

}

// Printing values with proper index.

printf("\nDisplaying values:\n");

for (int i = 0; i < 2; ++i)

{

for (int j = 0; j < 3; ++j)

{

for (int k = 0; k < 2; ++k)

{

printf("test[%d][%d][%d] = %d\n", i, j, k, test[i][j][k]);

}

}

}

return 0;

}

# Особености на многомерните масиви

Когато масив е двумерен и има еднакъв брой редове и колони, то той е квадратен масив и има главен диагонал и второстепенен диагонал.

Задачи:

Попълнете масив NxN елемента и го попълнете с числа.

* Изкарайте елементите на главния диагонал
* Изкарайте елементите на второстепенния диагонал
* Изведете елементите над главния диагонал
* Изведете елементите под главния диагонал

Направете програма, която проверява дали един квадрат е магически(сумите на числата във всички редове и колони да е равна на едно и също число).