Chapter 2 : Indexed variables

Part IV : Two Dimensional Array

Definition:

The two dimensional array (2D array) it can be viewed as an array of arrays. The 2D array is required in the case of stored data that takes tabular form or matrices.

Declaration :

Declaration of 2D array can be done by the following task

```
data type array_name[x][y];
```

Example:

long M[2][3];

	col 0	col 1	col 2
row 0	M[0][0]	M[0][1]	M[0][2]
row 1	M[1][0]	M[1][1]	M[1][2]

long: is the data type of the 2D array.

M: is the name of the 2D array.

2: between square brackets represents the number of rows in the 2D array.

3: is the number of columns in the 2D array.

Initializing 2D array:

There are many ways to initialize a 2D array the simple one is the initialization during declaration as

float X[3][3]={1.2, 5, 10, 2, 4, 3, 6, 9, 7};

Or in equivalent way as

float X[3][3]={{1.2, 5, 10}, {2, 4, 3}, {6, 9, 7}};

	col 0	col 1	col 2
row 0	X[0][0]=1.2	X[0][1]=5	X[0][2]=10
row 1	X[1][0]=2	X[1][1]=4	X[1][2]=3
row 2	X[2][0]=6	X[2][1]=9	X[2][2]=7

To assign values to elements in 2D array the scanf() can be used as follows

```
int M[2][2];
scanf("%d", &M[0][0]);/* to enter the value of the first element in array M*/
```

Printing 2D arrays:

To print all elements of 2D array nested for loop is used as follows:

Program

```
#include <stdio.h>

int main() {
  int M[3][4]={1,2,3,7,2,4,5,8,3,5,6,9};
  int i, j;

  for(i=0;i<3;i++)
  {
     for(j=0;j<4;j++)
     {
        printf("%d ", M[i][j]);
     }
     printf("\n");
  }


   return 0;
}
```

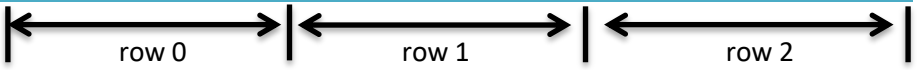
Output

```
1 2 3 7
2 4 5 8
3 5 6 9
```

Storage of 2D array:

The elements of 2D array are stored in memory in a linear way in row-major order i.e. the rows are placed one by one in memory as one dimensional array. The element of the 2D array int x[3][2]={1, 2, 2, 1, 1, 4}; it can be represented in this way

Element	x[0][0]	x[0][1]	x[1][0]	x[1][1]	x[2][0]	x[2][1]
Value	1	2	2	1	1	4
Address	2000	2004	2008	2012	2016	2020



Example:

Program	*Output*

#include <stdio.h>
 	
int main() {
  int mat[3][2]={{1,2},{5,9},{7,9}};
  int i,j;
  int* pt;
  pt=&mat[0][0];
 	
 for(i=0;i<6;i++)
 printf("%p\n", pt+i);
return 0;
}
	0x7ffe872ca670
	0x7ffe872ca674
	0x7ffe872ca678
	0x7ffe872ca67c
	0x7ffe872ca680
	0x7ffe872ca684

Length of 2D array:

The total number of elements in 2D array=number of rows × number of columns. For example the 2D array int A[10][7] can store 10×7=70 elements. Each of element occupy a space of 4 bytes in memory, and then the array has a storage space given with

size of A[10][7] =70×4=280 bytes.

Example:

Program	*Output*

#include <stdio.h>
 	
int main() {
  int mat[3][2]={{1,2},{5,9},{7,9}};
  int length;
 	
length=sizeof(mat)/sizeof(mat[0][0]);
  printf("length=%d", length);
return 0;
}
	length=6

Some operations on 2D arrays:

1- C program to find 0th column and 0th row of 2D array:

Program	*Output*

#include <stdio.h>
int main()
{
  int a[3][3]={{4,3,2},{5,4,1},{6,9,8}};
  int i, j;
  printf("elements of 0-column:\n");
  for(i=0;i<3;i++)
  {
  printf("%d\n", a[i][0]);
  }
  printf("\n----------\n");
  printf("elements of 0-row:\n");
  for(j=0;j<3;j++)
  {
  printf("%d\t", a[0][j]);
  }
  return 0;
}
	elements of 0-column:
	4
	5
	6
	
	---------- 	
	elements of 0-row:
	4 3 2

2- Program to find the position of an element in 2D array

Program	*Output*

#include <stdio.h>
int main()
{
int b[4][4]={{6,4,3,2},{10,5,4,1},{4,6,9,8},{9,8,7,12}};
int i, j, a, c=0;
printf("Enter the number you are looking for:\n");
scanf("%d", &a);
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
if(a==b[i][j])
{
printf("The number position is (%d, %d)\n", i, j);
c++;
}
}
}
if(c==0)
{
printf("Not found");
}
return 0;
}
	Enter the number you are looking for:
	4
	The number position is (0, 1)
	The number position is (1, 2)
	The number position is (2, 0)

3- C program to solve a system of equations with two variables using matrices:

a- The form of a system of equation can be written as

$$a_1x + b_1y = c_1$$
$$a_2x + b_2y = c_2$$

The theoretical solution of this system of equations is given by

$$x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

$$y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

Where $\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1 b_2 - a_2 b_1$ is the determinant of the matrix.

b- Next, we will write the C program that finds a solution of the following system of equations

$$2x + 4y = 1$$
$$x + 3y = 2$$

Program	*Output*

#include <stdio.h>
float determinanat(float a[2][2])
{ float det;
det=a[0][0]*a[1][1]-a[1][0]*a[0][1];
return det;}
int main(){
float b[2][2];
float b1[2][2];
float b2[2][2];
int i,j,k,l;
printf("Enter the elements of the matrix\n");
for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
{
scanf("%f",&b[i][j]);
 	
}
}
for(k=0;k<2;k++)
{
for(l=0;l<2;l++)
{
b1[k][l]=b[k][l];
b2[k][l]=b[k][l];
}}
float c1=1, c2=2, x, y, d, z1, z2;
d=determinanat(b);
printf("d=%f\n", d);
b1[0][0]=c1;
b1[1][0]=c2;
b2[0][1]=c1;
b2[1][1]=c2;
x=determinanat(b1)/d;
y=determinanat(b2)/d;
	Enter the elements of the matrix
	2
	4
	1
	3
	d=2.000000
	x=-2.500000
	y=1.500000
	solutions substitutions
	z1=0.000000
	z2=0.000000

```
printf("x=%f\ny=%f\n", x, y);
printf("solutions substitutions\n");
z1=2*x+4*y-1;
z2=x+3*y-2;
printf("z1=%f\nz2=%f", z1, z2);
return 0;}
```