Programming and Problem Solving through C Language
O Level / A Level

Chapter - 5 : Array

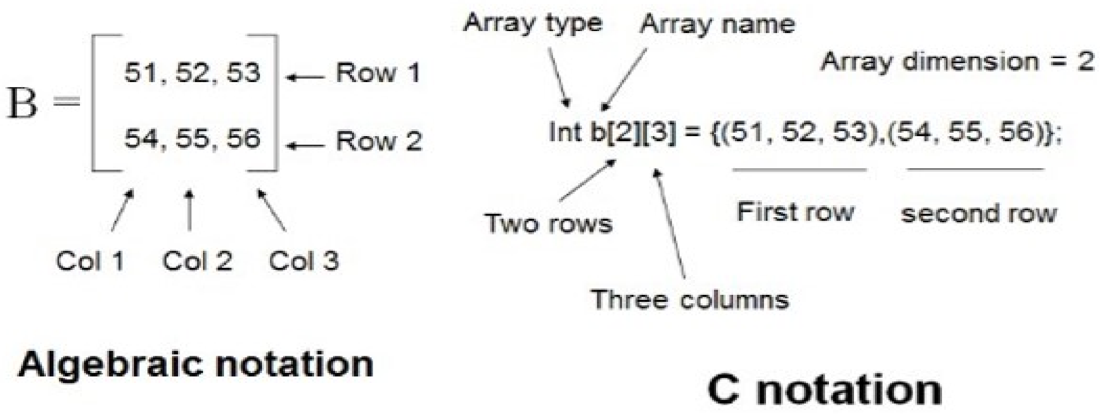
Multidimensional Arrays - Two-Dimensional Arrays

- A two-dimensional array has two subscripts.
- A two-dimensional array has a row-and-column structure as shown below:

  int matrix[row][column];

  Where row and column can only be an integer.

What is a Two-dimensional array?



Indexes in 2D arrays

- Assume that the two dimensional array called val is declared and looks like the following:

val	Col 0	Col 1	Col 2	Col 3
Row 0	8	16	9	52
Row 1	3	15	27	6
Row 2	14	25	2	10

- To access the cell containing 6, we reference val[1][3],that is, row 1, column 3.

DECLARATION

- How to declare a multidimensional array?
  - int b[2][3];
    - The name of the array to be b.
    - The type of the array elements to be int.
    - The dimension to be 2 (two pairs of brackets []).
    - The number of elements or size to be 2*3 = 6.

Initialization of a 2d array

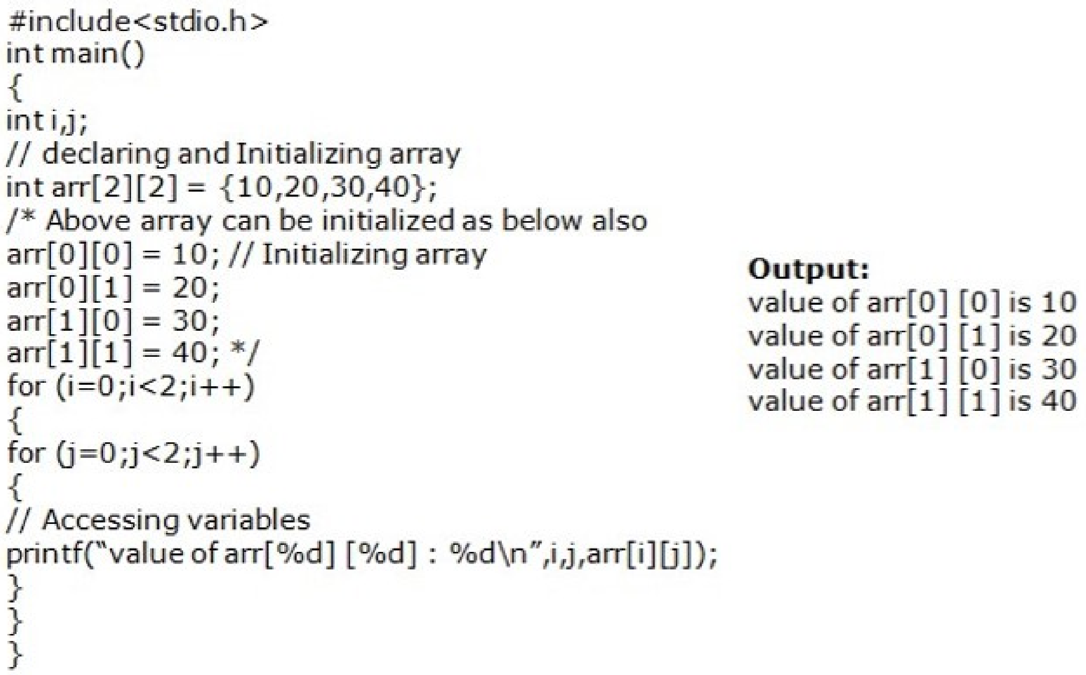
```
// Different ways to initialize two-dimensional array

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};
```

Initialization of a 3d array

```
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}}    };
```

Example: A program to input elements in a two dimensional array and print it

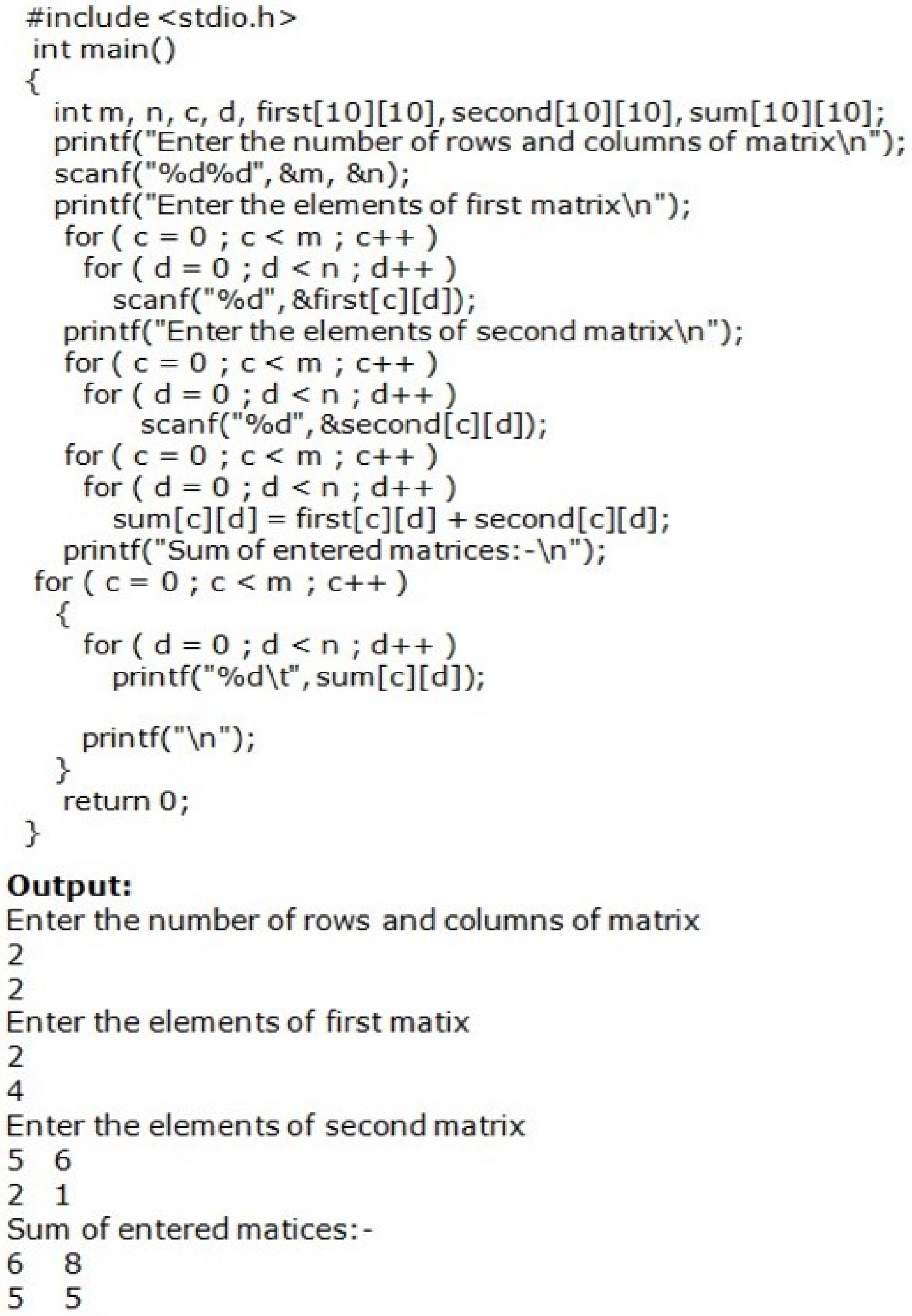


```
#include<stdio.h>
int main()
{
int i,j;
// declaring and Initializing array
int arr[2][2] = {10,20,30,40};
/* Above array can be initialized as below also
arr[0][0] = 10; // Initializing array
arr[0][1] = 20;
arr[1][0] = 30;
arr[1][1] = 40; */
for (i=0;i<2;i++)
{
for (j=0;j<2;j++)
{
// Accessing variables
printf(“value of arr[%d] [%d] : %d\n”,i,j,arr[i][j]);
}
}
}
```

Output:

```
value of arr[0] [0] is 10
value of arr[0] [1] is 20
value of arr[1] [0] is 30
value of arr[1] [1] is 40
```

Addition of Two Matrices: Program and Output



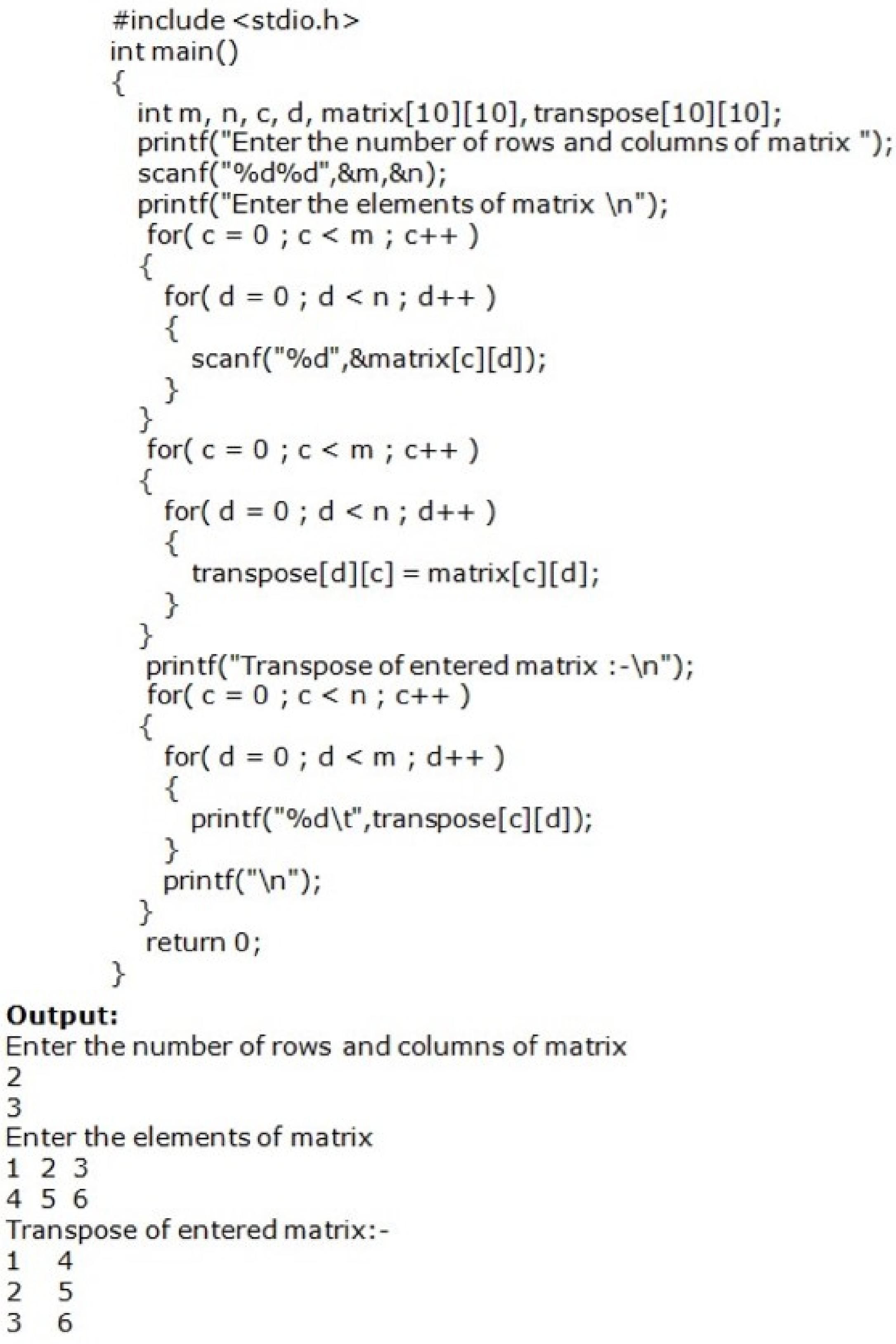
```
 #include <stdio.h>
  int main()
 {
    int m, n, c, d, first[10][10], second[10][10], sum[10][10];
    printf("Enter the number of rows and columns of matrix\n");
    scanf("%d%d", &m, &n);
    printf("Enter the elements of first matrix\n");
     for ( c = 0 ; c < m ; c++ )
       for ( d = 0 ; d < n ; d++ )
          scanf("%d", &first[c][d]);
     printf("Enter the elements of second matrix\n");
     for ( c = 0 ; c < m ; c++ )
       for ( d = 0 ; d < n ; d++ )
             scanf("%d", &second[c][d]);
     for ( c = 0 ; c < m ; c++ )
       for ( d = 0 ; d < n ; d++ )
          sum[c][d] = first[c][d] + second[c][d];
     printf("Sum of entered matrices:-\n");
  for ( c = 0 ; c < m ; c++ )
    {
       for ( d = 0 ; d < n ; d++ )
          printf("%d\t", sum[c][d]);

       printf("\n");
    }
     return 0;
 }
```

Output:

```
Enter the number of rows and columns of matrix
2
2
Enter the elements of first matix
2
4
Enter the elements of second matrix
5   6
2   1
Sum of entered matices:-
6    8
5    5
```

Transpose of Matrix

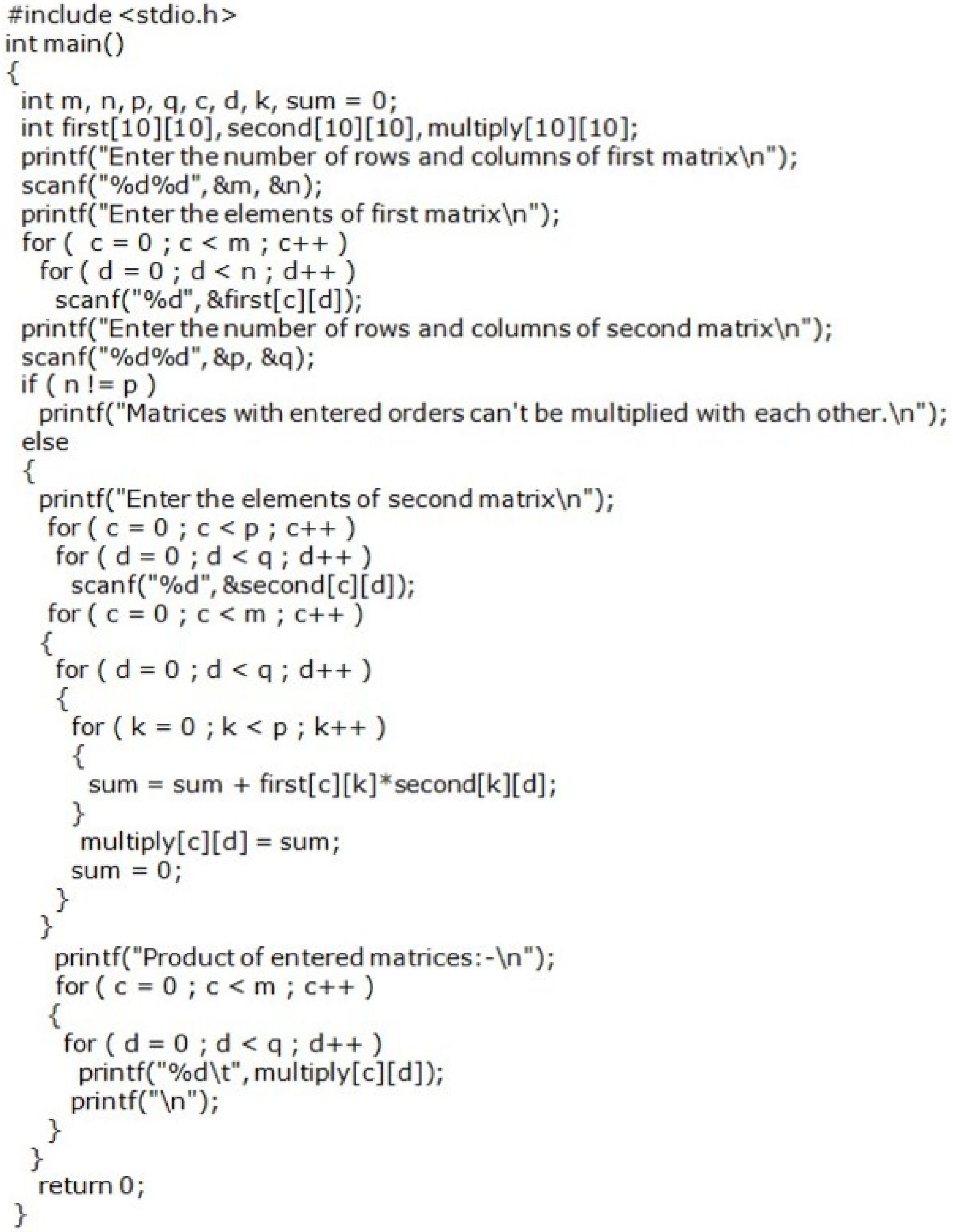


```
#include <stdio.h>
int main()
{
   int m, n, c, d, matrix[10][10], transpose[10][10];
   printf("Enter the number of rows and columns of matrix ");
   scanf("%d%d",&m,&n);
   printf("Enter the elements of matrix \n");
    for( c = 0 ; c < m ; c++ )
   {
     for( d = 0 ; d < n ; d++ )
     {
        scanf("%d",&matrix[c][d]);
     }
   }
    for( c = 0 ; c < m ; c++ )
   {
     for( d = 0 ; d < n ; d++ )
     {
        transpose[d][c] = matrix[c][d];
     }
   }
    printf("Transpose of entered matrix :-\n");
    for( c = 0 ; c < n ; c++ )
   {
     for( d = 0 ; d < m ; d++ )
     {
        printf("%d\t",transpose[c][d]);
     }
     printf("\n");
   }
    return 0;
}
```

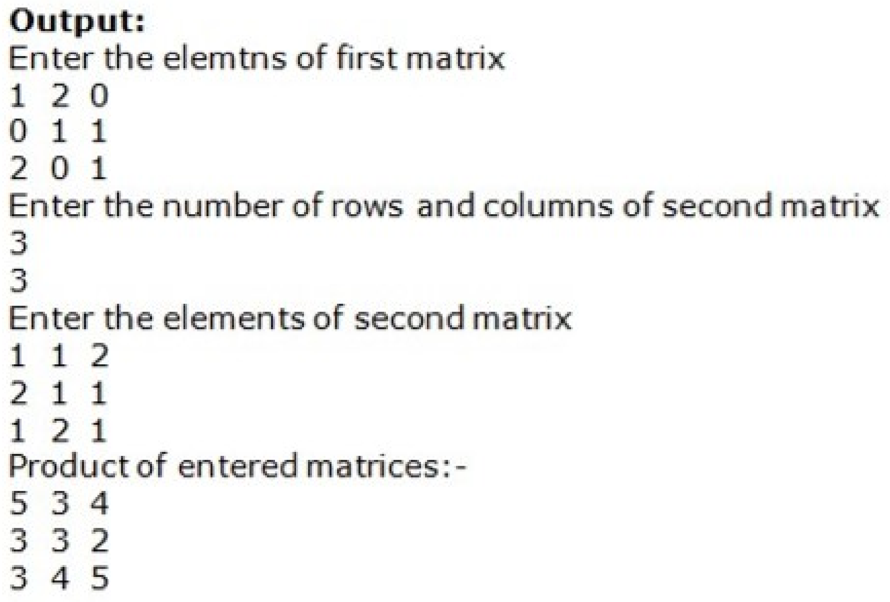
Output:

```
Enter the number of rows and columns of matrix
2
3
Enter the elements of matrix
1 2 3
4 5 6
Transpose of entered matrix:-
1    4
2    5
3    6
```

Multiplication of Two Matrices: Program and Output



```
#include <stdio.h>
int main()
{
  int m, n, p, q, c, d, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];
  printf("Enter the number of rows and columns of first matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter the elements of first matrix\n");
  for (  c = 0 ; c < m ; c++ )
    for ( d = 0 ; d < n ; d++ )
      scanf("%d", &first[c][d]);
  printf("Enter the number of rows and columns of second matrix\n");
  scanf("%d%d", &p, &q);
  if ( n != p )
    printf("Matrices with entered orders can't be multiplied with each other.\n");
  else
  {
    printf("Enter the elements of second matrix\n");
     for ( c = 0 ; c < p ; c++ )
      for ( d = 0 ; d < q ; d++ )
        scanf("%d", &second[c][d]);
     for ( c = 0 ; c < m ; c++ )
    {
      for ( d = 0 ; d < q ; d++ )
      {
        for ( k = 0 ; k < p ; k++ )
        {
          sum = sum + first[c][k]*second[k][d];
        }
         multiply[c][d] = sum;
        sum = 0;
      }
    }
      printf("Product of entered matrices:-\n");
      for ( c = 0 ; c < m ; c++ )
     {
       for ( d = 0 ; d < q ; d++ )
         printf("%d\t", multiply[c][d]);
        printf("\n");
     }
   }
    return 0;
 }
```



Output:

```
Enter the elemtns of first matrix
1 2 0
0 1 1
2 0 1
Enter the number of rows and columns of second matrix
3
3
Enter the elements of second matrix
1 1 2
2 1 1
1 2 1
Product of entered matrices:-
5 3 4
3 3 2
3 4 5
```

Assignment

1. Write a program to read an array of dimension 3 X 3, find and print the maximum and minimum of this array.
2. Write a program to read a Matrix of dimension 3 X 3, find the sum of diagonal items.
3. Write a program to read a Matrix of dimension 3 X 3, find the sum of all elements.