

Exercise 1 – Write a program to read rows & columns of two matrices **A** and **B** in (**r1**, **c1**) and (**r2**, **c2**), respectively. If they have same dimension (i.e., they must have the same number of rows and columns), then compute their sum and display all three matrices **A**, **B** and the **resultant**. An example would be as follows:

```
Enter no. of rows & columns of matrix A: 2 3
Enter no. of rows & columns of matrix B: 2 3

Enter data in matrix A
1 2 3 4 5 6
Enter data in matrix B
1 1 1 1 1 1

Matrix A
1 2 3
4 5 6
Matrix B
1 1 1
1 1 1
Resultant matrix
2 3 4
5 6 7
```

Program –

```
#include<stdio.h>

//Function to add matrix
void matrix_adding(int m, int n, int p, int q)
{
    int i,j,A[100][100],r,c,x,y,B[100][100],C[100][100];
    printf("\nEnter data in matrix A\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            scanf("%d",&A[i][j]);
        }
    }
    printf("Enter data in matrix B\n");
    for(x=0;x<p;x++)
    {
        for(y=0;y<q;y++)
        {
            scanf("%d",&B[x][y]);
        }
    }
}
```

```
printf("\nMatrix A\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
    {
        printf("%d ",A[i][j]);
    }
    printf("\n");
}
printf("Matrix B\n");
for(x=0;x<p;x++)
{
    for(y=0;y<q;y++)
    {
        printf("%d ",B[x][y]);
    }
    printf("\n");
}
printf("Resultant matrix \n");
for(r=0;r<p;r++)
{
    for(c=0;c<q;c++)
    {
        C[r][c] = A[r][c] + B[r][c];
        printf("%d ",C[r][c]);
    }
    printf("\n");
}
}

int main()
{
    int m,n,p,q;
    printf("Enter no. of rows & columns of matrix A: ");
    scanf("%d%d",&m,&n);
    printf("Enter no. of rows & columns of matrix B: ");
    scanf("%d%d",&p,&q);
    if(m==p && n==q)
    {
        matrix_adding(m,n,p,q);    //Function Calling
    }
    else
    {
        printf("Error! Unequal Dimensions...");
    }
    return 0;
}
```

Output –

```
Enter no. of rows & columns of matrix A: 2 3
Enter no. of rows & columns of matrix B: 2 3
```

```
Enter data in matrix A
1 2 3 4 5 6
Enter data in matrix B
1 1 1 1 1 1
```

```
Matrix A
1 2 3
4 5 6
Matrix B
1 1 1
1 1 1
Resultant matrix
2 3 4
5 6 7
```

Exercise 2 – Write a program to read matrix *A* and after transpose it into matrix *B* print both the matrices. An example would be as follows:

```
Enter no. of rows & columns of matrix A: 2 3
Enter data in matrix A
1 2 3 4 5 6

Matrix A
1 2 3
4 5 6
Transposed Matrix B
1 4
2 5
3 6
```

Program –

```
#include<stdio.h>

//Function to Transpose a Matrix
void transpose(int p, int q)
{
    int i,j,A[100][100];
    printf("Enter data in matrix A\n");
    for(i=0;i<p;i++)
    {
        for(j=0;j<q;j++)
        {
```

```
        scanf("%d",&A[i][j]);
    }
}
printf("\nMatrix A\n");
for(i=0;i<p;i++)
{
    for(j=0;j<q;j++)
    {
        printf("%d ",A[i][j]);
    }
    printf("\n");
}
printf("Transposed Matrix B\n");
for(i=0;i<q;i++)
{
    for(j=0;j<p;j++)
    {
        printf("%d ",A[j][i]);
    }
    printf("\n");
}
}

int main()
{
    int p,q;
    printf("Enter no. of rows & columns of matrix A: ");
    scanf("%d%d",&p,&q);
    transpose(p,q);
    return 0;
}
```

Output –

```
Enter no. of rows & columns of matrix A: 2 3
Enter data in matrix A
1 2 3 4 5 6

Matrix A
1 2 3
4 5 6
Transposed Matrix B
1 4
2 5
3 6
```