***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

***Exercise 3–*** *Write a program to read rows & columns of two matrices* ***A*** *and* ***B*** *in* ***(r1, c1)*** *and* ***(r2, c2)****, respectively. If the number of columns in* ***A*** *must be equal the number of rows in* ***B*** *i.e., if* ***A*** *is an* ***r1* x *c1*** *matrix and* ***B*** *is an* ***r2* x *c2*** *matrix,* ***c1 = r2****, then write the following functions to compute it’s corresponding job.*

|  |  |
| --- | --- |
| *Function* | *Description* |
| *read(int [][10],int,int );* | *Read data from A & B matrices* |
| *write(int [][10],int,int );* | *Print A, B & resultant matrices* |
| *multi(int [][10], int [][10], int [][10], int,int,int);* | *Multiply both A & B matrices and stored it into resultant matrix* |

*An example would be as follows:*

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

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

Enter data in matrix A

8 1 2 -5 6 7

Enter data in matrix B

-5 1 0 2 -11 7

Matrix A

8 1 2

-5 6 7

Matrix B

-5 1

0 2

-11 7

Resultant matrix

-62 24

-52 56

*Program –*

#include<stdio.h>

//Read data from A & B matrices

void read(int P[][10], int r, int c)

{

int i,j;

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

{

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

{

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

}

}

}

//Print A, B & resultant matrices

void write(int P[][10], int r, int c)

{

int i,j;

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

{

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

{

printf("%d\t",P[i][j]);

}

printf("\n");

}

}

//Multiply both A & B matrices and stored it into resultant matrix

void multi(int A[][10], int B[][10], int C[][10], int m, int q, int p)

{

int i,j,k,sum=0;

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

{

for(j=0;j<q;j++)

{

for(k=0;k<p;k++)

{

sum += A[i][k] \* B[k][j];

}

C[i][j] = sum;

sum = 0;

}

}

}

int main()

{

int m,n,p,q,A[10][10],B[10][10],C[10][10];

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(n==p)

{

printf("\nEnter data in matrix A\n");

read(A,m,n);

printf("Enter data in matrix B\n");

read(B,p,q);

printf("\nMatrix A\n");

write(A,m,n);

printf("Matrix B\n");

write(B,p,q);

printf("Resultant matrix\n");

multi(A,B,C,m,q,p);

write(C,m,q);

}

else

{

printf("Error! Unequal Dimensions (Columns of Matrix A != Rows of Matrix B)");

}

return 0;

}

*Output –*

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

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

Enter data in matrix A

8 1 2 -5 6 7

Enter data in matrix B

-5 1 0 2 -11 7

Matrix A

8 1 2

-5 6 7

Matrix B

-5 1

0 2

-11 7

Resultant matrix

-62 24

-52 56