**ASSIGNMENT – 16**

**1. Write a program to calculate the sum of two matrices each of order 3x3.**

#include<stdio.h>

int main ()

{

    int a [3][3], b [3][3], c [3][3];

    printf ("Enter 1st matrix: ");

    for (int i=0; i<3; i++)

    {

        for (int j=0; j<3; j++)

        {

            scanf ("%d", &a[i][j]);

        }

    }

    printf ("Enter 2nd matrix: ");

    for (int i=0; i<3; i++)

    {

        for (int j=0; j<3; j++)

        {

            scanf ("%d", &b[i][j]);

        }

    }

    for (int i=0; i<3; i++)

    {

        for (int j=0; j<3; j++)

        {

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

        }

    }

    printf ("Sum of matrix is:");

    for (int i=0; i<3; i++)

    {

        for (int j=0; j<3; j++)

        {

            printf ("%d ", c[i][j]);

        }

    }

    return 0;

}

**2. Write a program to calculate the product of two matrices each of order 3x3.**

#include <stdio.h>

int main ()

{

    int a [3][3], b [3][3];

    printf ("Enter 1st array: ");

    for (int i = 0; i < 3; i++)

    {

        for (int j = 0; j < 3; j++)

        {

            scanf ("%d", &a[i][j]);

        }

    }

    printf ("Enter 2nd array: ");

    for (int i = 0; i < 3; i++)

    {

        for (int j = 0; j < 3; j++)

        {

            Scanf ("%d", &b[i][j]);

        }

    }

    // finding sum

    for (int k = 0; k < 3; k++)

    {

        for (int i = 0; i < 3; i++)

        {

            int sum = 0;

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

            {

                sum += a[i][j] \* b[i][j];

            }

            printf ("%d  ", sum);

        }

        printf("\n");

    }

    return 0;

}

**3. Write a program in C to find the transpose of a given matrix.**

#include<stdio.h>

int main ()

{

    int n;

    printf ("Enter size of matrix: ");

    scanf ("%d", &n);

    int arr [n][n];

    printf ("Enter array of %d X %d: ", n, n);

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            scanf ("%d", &arr[i][j]);

        }

    }

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            int temp = 0;

            if (i > j)

            {

               temp = arr[i][j];

               arr[i][j] = arr[j][i];

               arr[j][i] = temp;

            }

        }

    }

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            printf ("%d ", arr[i][j]);

        }

    }

    return 0;

}

**4. Write a program in C to find the sum of right diagonals of a matrix.**

#include <stdio.h>

int main ()

{

    int n, sum = 0;

    printf ("Enter the size of matrix: ");

    scanf ("%d", &n);

    int arr[n][n];

    printf ("Enter elements of %d x %d matrix: ", n, n);

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < n; j++)

        {

            scanf ("%d", &arr[i][j]);

        }

    }

    // finding sum of right diagonal

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < n; j++)

        {

            if (j == n - i - 1)

            {

                sum += arr[i][j];

            }

        }

        // printf ("i=%d j=%d ", i, j);

    }

    // printing sum

    printf ("Sum of right diagonal elements of %d x %d matrix is: %d", n, n, sum);

return 0;

}

**5. Write a program in C to find the sum of left diagonals of a matrix.**

#include <stdio.h>

int main ()

{

    int n, sum = 0;

    printf ("Enter size of matrix: ");

    scanf ("%d", &n);

    int arr[n][n];

    printf ("Enter elements of %d x %d matrix: ");

    // Taking input

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < n; j++)

        {

            scanf ("%d", &arr[i][j]);

        }

    }

    // Finding sum...

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            if (i==j)

            {

                sum += arr[i][j];

            }

        }

    }

    // Print sum

    printf ("Sum of right diagonal elements is: %d", sum);

    return 0;

}

**6. Write a program in C to find the sum of rows and columns of a Matrix.**

// Write a program in C to find the sum of rows and columns of a Matrix

#include<stdio.h>

int main ()

{

    int n, sor = 0, soc = 0;

    printf ("Enter size of matrix: ");

    scanf ("%d", &n);

    int arr[n][n];

    printf ("Enter the elements of matrix: ");

    // Taking input

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            scanf ("%d", &arr[i][j]);

        }

    }

    //Finding sum of rows

    for (int i=0; i<n; i++)

    {

        sor = 0;

        for (int j=0; j<n; j++)

        {

            sor += arr[i][j];

        }

        printf ("Sum of row %d = %d\n", i, sor);

    }

    // Finding sum of columns

    for (int i=0; i<n; i++)

    {

        soc = 0;

        for (int j=0; j<n; j++)

        {

            soc += arr[j][i];

        }

        printf ("Sum of %d th column: %d\n", i, soc);

    }

    return 0;

}

**7. Write a program in C to print or display the lower triangular of a given matrix.**

#include<stdio.h>

int main ()

{

    int n;

    printf ("Enter size of matrix: ");

    scanf ("%d", &n);

    int arr [n][n];

    printf ("Enter elements in %d x %d matrix: ", n, n);

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            scanf ("%d", &arr[i][j]);

        }

    }

    // Displaying lower triangular

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            if(i>=j)

                printf ("%d ", arr[i][j]);

            else

                printf ("\0");

        }

        printf ("\n");

    }

    return 0;

}

**8. Write a program in C to print or display an upper triangular matrix.**

#include<stdio.h>

int main ()

{

    int n;

    printf ("Enter size of array:");

    scanf ("%d", &n);

    int arr[n][n];

    printf ("Enter elements in %d x %d array:", n, n);

    // Taking input

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            scanf ("%d", &arr[i][j]);

        }

    }

    // printing Upper triangular matrix

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            if (i<=j)

                printf ("%d ", arr[i][j]);

            else

                printf ("\0");

        }

        printf ("\n");

    }

    return 0;

}

**9. Write a program in C to accept a matrix and determine whether it is a sparse matrix.**

// Write a program in C to accept a matrix and determine whether

//it is a sparse matrix.

#include<stdio.h>

int main ()

{

    int n;

    printf ("Enter size of matrix: ");

    scanf ("%d", &n);

    int arr[n][n];

    printf ("Enter %d elements in the array: \n",n\*n);

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            scanf ("%d", &arr[i][j]);

        }

    }

    // Showing the matrix

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

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

        }

        printf ("\n");

    }

    int cnt = 0;

    for (int i=0; i<n; i++)

    {

        for (int j = 0; j<n; j++)

        {

            if (arr[i][j] == 0)

                cnt++;

        }

    }

    if (cnt > 4)

        printf ("it's a sparse Matrix\n");

    else

        printf ("Not a sparse matrix\n");

        return 0;

}

**10. Write a program in C to find the row with maximum number of 1s.**

#include<stdio.h>

#include<stdlib.h>

int main ()

{

    int n;

    printf ("Enter size of array: ");

    scanf ("%d", &n);

    int arr[n][n];

    printf ("Enter %d X %d numbers: ", n, n);

    // Taking input

    for (int i=0; i<n; i++)

    {

        for (int j=0; j<n; j++)

        {

            scanf ("%d", &arr[i][j]);

        }

    }

    // Finding count of 1

    int cnt\_1[n];

    for (int i=0; i<n; i++)

    {

        int cnt=0;

        for (int j=0; j<n; j++)

        {

            if (arr[i][j] == 1)

                cnt++;

        }

        cnt\_1 [i] = cnt;

        continue;

    }

    // Find Maximum

    int max =-1;

    for (int i=0; i<n; i++)

    {

        if (cnt\_1[i]> max)

            max = cnt\_1[i];

    }

    // Finding row with maximum 1

    for (int i=0; i<n; i++)

    {

        if (cnt\_1[i] == max)

        {

            printf ("Row with maximum 1 is %d", i);

        }

    }

    return 0;

}