**Dsa lab 2**

1. **Write a C program to store the elements in a 2D array and display it &represent it in row major order & display it.**

**Code**

#include <stdio.h>

int main(){

    int m , n;

    printf("Enter the total number of rows : ");

    scanf("%d",&m);

    printf("Enter the total number of column : ");

    scanf("%d",&n);

    int arr[m][n];

    printf("Enter all the elements : ");

    for (int i = 0; i < m; i++){

        for (int j = 0; j < n; j++){

            scanf("%d",&arr[i][j]);

        }

    }

    printf("Row-Major Order: ");

    for (int i = 0; i < m; i++){

        for (int j = 0; j < n; j++){

            printf("%d ", arr[i][j]);

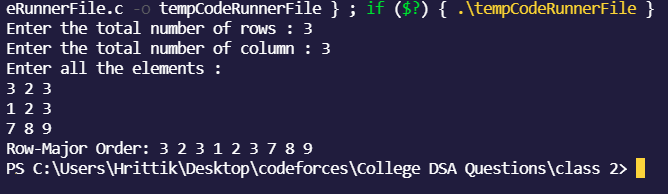
        }

    }

    return 0;

}

Output



1. **Write a C program to test a given matrix is sparse or not. If it is sparse then represent it as 3-tuple format.**

**code**

#include <stdio.h>

#define srow 50

#define mrow 20

#define mcolumn 20

*/\*Begin of main\*/*

int main()

{

    int mat[mrow][mcolumn], sparse[srow][3];

    int i, j, nzero = 0, zero = 0, mr, mc, sr, s;

*//taking inputs*

    printf("Enter number of rows : ");

    scanf("%d", &mr);

    printf("Enter number of columns : ");

    scanf("%d", &mc);

    for (i = 0; i < mr; i++)

        for (j = 0; j < mc; j++)

        {

*//taking inputs of rows and columns*

            printf("Enter element for row %d,column %d : ", i + 1, j + 1);

            scanf("%d", &mat[i][j]);

        }

*//printing entered matrix*

    printf("Entered matrix is : \n");

    for (i = 0; i < mr; i++)

    {

        for (j = 0; j < mc; j++)

        {

            printf("%6d", mat[i][j]);

            if (mat[i][j] != 0)

            {

                nzero++;

            }

            else

            {

                zero++;

            }

        }

        printf("\n");

    }

    if (nzero < zero)

    {

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

    }

    else

    {

        printf("It is a sparse matrix \n");

        sr = nzero + 1;

        sparse[0][0] = mr;

        sparse[0][1] = mc;

        sparse[0][2] = nzero;

        s = 1;

        for (i = 0; i < mr; i++)

            for (j = 0; j < mc; j++)

            {

                if (mat[i][j] != 0)

                {

                    sparse[s][0] = i + 1;

                    sparse[s][1] = j + 1;

                    sparse[s][2] = mat[i][j];

                    s++;

                }

            }

*//printing sparse matrix*

        printf("Sparse matrix is :\n");

        for (i = 0; i < sr; i++)

        {

            for (j = 0; j < 3; j++)

                printf("%5d", sparse[i][j]);

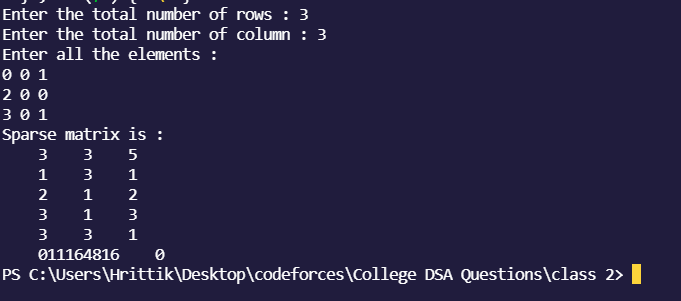
            printf("\n");

        }

    }

}

**Output**



1. **Write a C program to find the transpose of a sparse matrix.**

**code**

#include <stdio.h>

int main()

{

    int a[10][10], transpose[10][10], r, c;

    printf("Enter rows and columns: ");

    scanf("%d %d", &r, &c);

*// asssigning elements to the matrix*

    printf("\nEnter matrix elements:\n");

    for (int i = 0; i < r; ++i)

        for (int j = 0; j < c; ++j)

        {

            printf("Enter element a%d%d: ", i + 1, j + 1);

            scanf("%d", &a[i][j]);

        }

    int count0 = 0, countNon0 = 0;

*// printing the matrix a[][]*

    printf("\nEntered matrix: \n");

    for (int i = 0; i < r; ++i)

        for (int j = 0; j < c; ++j)

        {

            printf("%d  ", a[i][j]);

            if (a[i][j] == 0)

            {

                count0++;

            }

            else

            {

                countNon0++;

            }

            if (j == c - 1)

                printf("\n");

        }

    if (count0 > countNon0)

    {

        printf("It is a sparse matrix \n");

*// computing the transpose*

        for (int i = 0; i < r; ++i)

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

            {

                transpose[j][i] = a[i][j];

            }

*// printing the transpose*

        printf("\nTranspose of the matrix:\n");

        for (int i = 0; i < c; ++i)

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

            {

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

                if (j == r - 1)

                    printf("\n");

            }

    }

    else

    {

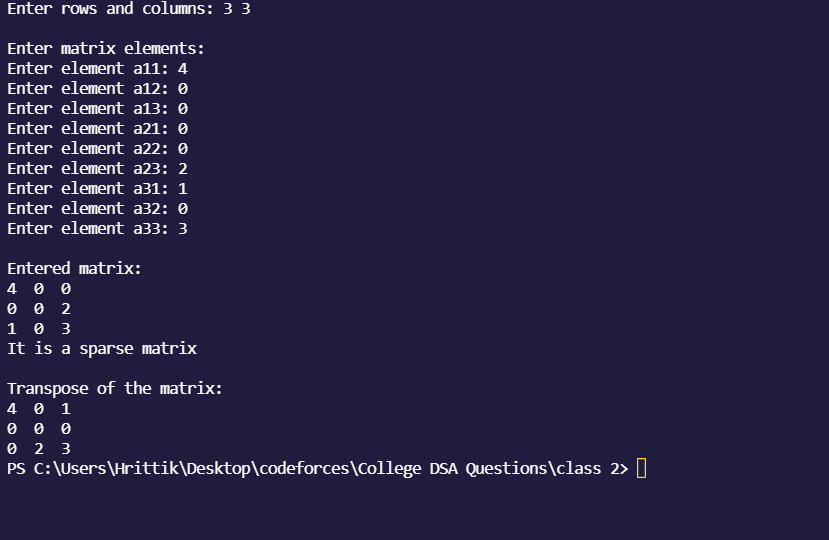
        printf("It is not a sparse matrix");

    }

    return 0;

}

**Output**



1. **Write a C program to find the matrix multiplication of two matrices.**

**code**

1. */\*\**
2. *\* Write a C program to find the matrix multiplication of two matrices.*
3. *\*/*
4. #include <stdio.h>
5. #include <stdlib.h>
6. int main()
7. {
8. int a[10][10], b[10][10], mul[10][10], r, c, i, j, k;
9. system("cls");
10. printf("enter the number of row=");
11. scanf("%d", &r);
12. printf("enter the number of column=");
13. scanf("%d", &c);
14. printf("enter the first matrix element=\n");
15. for (i = 0; i < r; i++)
16. {
17. for (j = 0; j < c; j++)
18. {
19. scanf("%d", &a[i][j]);
20. }
21. }
22. printf("enter the second matrix element=\n");
23. for (i = 0; i < r; i++)
24. {
25. for (j = 0; j < c; j++)
26. {
27. scanf("%d", &b[i][j]);
28. }
29. }
30. printf("multiply of the matrix=\n");
31. for (i = 0; i < r; i++)
32. {
33. for (j = 0; j < c; j++)
34. {
35. mul[i][j] = 0;
36. for (k = 0; k < c; k++)
37. {
38. mul[i][j] += a[i][k] \* b[k][j];
39. }
40. }
41. }
42. *//for printing result*
43. for (i = 0; i < r; i++)
44. {
45. for (j = 0; j < c; j++)
46. {
47. printf("%d\t", mul[i][j]);
48. }
49. printf("\n");
50. }
51. return 0;
52. }

**Output**

