

Maharaja Surajmal Institute

Affiliated to GGSIPU & NAAC 'A' grade accredited



DEPARTMENT OF COMPUTER APPLICATIONS

DATABASE MANAGEMENT SYSTEM

PRACTICAL FILE

SUBJECT CODE – BCA

SUBMITTED BY:

Name: Aman Tripathi

Enroll No: 05221202021

Sem: 2nd Sec: A (2nd shift)

SUBMITTED TO:

Mrs. Tarunim Sharma

Assistant Professor

INDEX

S.No.	Practical	Date	Sign
1.	WAP to find greatest among 3 numbers using a conditional operator.	21/04/2022	
2.	WAP to print a table of a number using a do-while loop.	21/04/2022	
3.	WAP to calculate the factorial of a number.	21/04/2022	
4.	WAP to print Fibonacci series using function.	21/04/22	
5.	WAP to enter the given list of numbers and find how many are positive, negative or zero.	26/04/2022	
6.	WAP to sort a given list of no in ascending order and print the original and sorted list using the function.	26/04/2022	
7.	WAP to calculate addition, subtraction, and multiplication of matrices.	26/04/2022	
8.	WAP to find the largest/smallest element of a matrix.		
9.	WAP to calculate the sum of each row and column and the total of all elements of the matrix.	28/04/2022	
10.	WAP to search elements from an array using linear search.	28/04/2022	
11.	WAP to search elements from an array using binary search.	28/04/2022	
12.	WAP to check whether numbers are a palindrome.	28/04/2022	
13.	WAP to calculate the sum of digits of the number.	28/04/2022	
14.	WAP for Bubble Sort to Sort Elements in An Order.	28/04/2022	
15.	WAP for insertion Sort to Sort Elements in An Order.	04/05/2022	
16.	WAP to for selection Sort to Sort Elements in An Order.	04/05/2022	
17.	WAP to mergeSort to Sort Elements in An Order.	05/05/2022	
18.	WAP to find lower and upper triangular matrices.	05/05/2022	
19.	WAP to check whether an entered is lower triangular.	05/05/2022	

Practical 1

Code:-

```
/ Q 1. WAP to find greatest among 3 number using conditional operator.

#include <stdio.h>

void main()
{
    int num1 = 0;
    int num2 = 0;
    int num3 = 0;
    int max = 0;

    // input from user
    printf("Enter the value of num1 : ");
    scanf("%d", &num1);
    printf("Enter the value of num2 : ");
    scanf("%d", &num2);
    printf("Enter the value of num3 : ");
    scanf("%d", &num3);

    // logic for program
    max = (num1 > num2)
        ? (num1 > num3 ? num1 : num3)
        : (num2 > num3 ? num2 : num3);

    // Output of Program
    printf("Maximum Number among three : %d", max);
```

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

Output:-

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS C:\Users\Aman Tripathi\OneDrive\Desktop\DSA file> cd "c:\U  
Enter the value of num1 : 23  
Enter the value of num2 : 65  
Enter the value of num3 : 11  
Maximum Number among three : 65
```

Practical 2

Code:-

```
/*
Q 2. WAP to print table of a number using do while loop.
*/

#include <stdio.h>

void main()
{
    int num = 0;

    // User Input
    printf("Enter the Number for table : ");

    scanf("%d", &num);

    // Logic
    int i = 1;
    int temp = 0;

    do
    {
        temp = i * num;

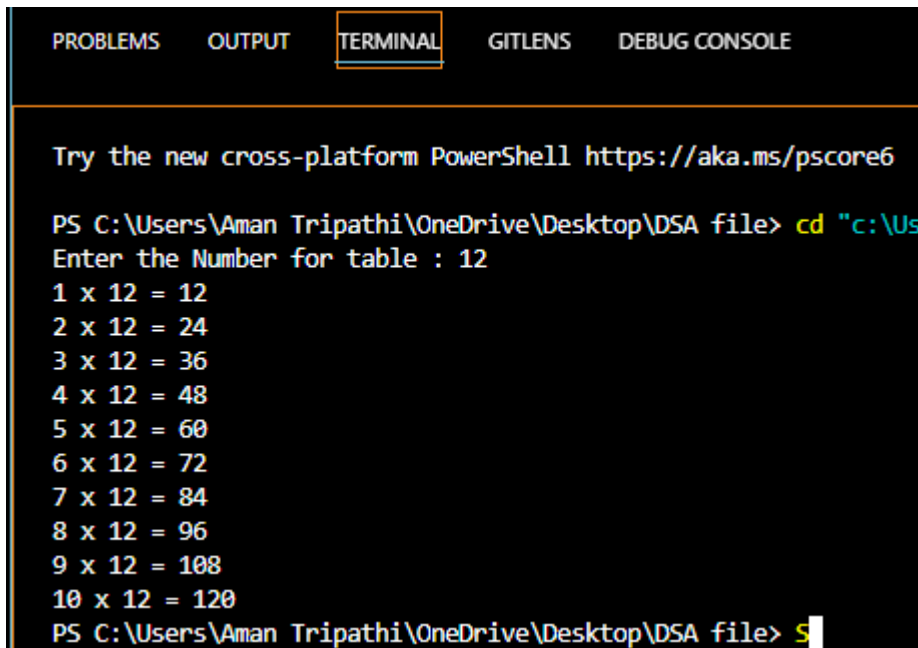
        printf("%d x %d = %d", i, num, temp);

        printf("\n");

        i++;
    } while (i <= 10);

    temp = 0;
}
```

Output:-



```
PROBLEMS  OUTPUT  TERMINAL  GITLENS  DEBUG CONSOLE

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Aman Tripathi\OneDrive\Desktop\DSA file> cd "c:\Us
Enter the Number for table : 12
1 x 12 = 12
2 x 12 = 24
3 x 12 = 36
4 x 12 = 48
5 x 12 = 60
6 x 12 = 72
7 x 12 = 84
8 x 12 = 96
9 x 12 = 108
10 x 12 = 120
PS C:\Users\Aman Tripathi\OneDrive\Desktop\DSA file> S
```

Practical 3

Code:-

```
// Q 3. WAP to calculate the factorial of a number.
#include <stdio.h>

int fact(int num)
{
    if (num == 0)
        return 1;
    else
        return num * fact(num - 1);
}

void main()
{
    int num = 0;

    printf("Enter the Number to find Factorial : ");

    scanf("%d", &num);

    int fac = fact(num);

    printf("%d", fac);
}
```

Output:-

```
PS C:\Users\Aman Tripathi\OneDrive\Desktop>
Enter the Number to find Factorial : 4
24
```


Practical 4

Code:-

```
// Q 4. WAP to print Fibonacci series using function.
// 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ...
#include <stdio.h>

void fibonacci(int range)
{
    int a = 0, b = 1, c;
    while (a <= range)
    {
        printf("%d\t", a);
        c = a + b;
        a = b;
        b = c;
    }
}

void main()
{
    int limit = 0;
    printf("Enter the Limit of the series : ");
    scanf("%d", &limit);
    printf("The fibonacci series is : ");
    fibonacci(limit);
}
```

Output:-

```
Enter the Limit of the series : 42
The fibonacci series is : 0    1    1    2    3    5    8    13    21    34
```

Practical 5

Code:-

```
/*
Q 5. WAP to enter given list of numbers and find how many
positive, negative or zero.
*/
#include <stdio.h>

int main()
{
    double num;

    printf("Enter a number: ");

    scanf("%lf", &num);

    if (num < 0.0)
        printf("You entered a negative number.");
    else if (num > 0.0)
        printf("You entered a positive number.");
    else
        printf("You entered 0.");

    return 0;
}
```

Output:-

```
Enter a number: 5
You entered a positive number.
```

```
Enter a number: 0
You entered 0.
```

```
Enter a number: -1
You entered a negative number.
```

Practical 6

Code:-

```
// Q 6. WAP to sort a given list of no in ascending order and print
#include <stdio.h>

// Sort the array in ascending order
void sort(int arr[], int length)
{
    int temp = 0;
    for (int i = 0; i < length; i++)
    {
        for (int j = i + 1; j < length; j++)
        {
            if (arr[i] > arr[j])
            {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int main()
{
    // Initialize array
    int n = 0;

    printf("Enter the Size of array : ");

    scanf("%d", &n);
```

```

int arr[n];

printf("Enter the Element of the Array: ");

for (int i = 0; i < n; i++)

{ scanf("%d", &arr[i]); }

// Calculate length of array arr

int length = sizeof(arr) / sizeof(arr[0]);

// Displaying elements of original array

printf("Elements of original array: \n");

for (int i = 0; i < length; i++)

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

sort(arr, length);

printf("\n");

// Displaying elements of array after sorting

printf("Elements of array sorted in ascending order: \n");

for (int i = 0; i < length; i++)

{

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

} return 0;

}

```

Output:-

```

Enter the Size of array : 5
Enter the Element of the Array: 2
1
3
4
5
Elements of original array:
2 1 3 4 5
Elements of array sorted in ascending order:
1 2 3 4 5

```

Practical 7

Code:-

```
/*Q 7. WAP to calculate addition, subtraction, multiplication of
matrix.*/

#include <stdio.h>

#include <stdlib.h>

int main()

{

    int a[10][10], b[10][10], mul[10][10], r, c, i, j, k;

    system("cls");

    printf("enter the number of row : ");

    scanf("%d", &r);

    printf("enter the number of column : ");

    scanf("%d", &c);

    printf("enter the first matrix element : \n");

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

    {

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

        {

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

        }

    }

    printf("enter the second matrix element : \n");

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

    {

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

        {
```

```

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

    }

}

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

for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
    {
        mul[i][j] = 0;

        for (k = 0; k < c; k++)
        {
            mul[i][j] += a[i][k] * b[k][j];
        }
    }
}

// for printing result
for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
    {
        printf("%d\t", mul[i][j]);
    }

    printf("\n");
}

// Addition of Matrix
printf("Addition of the matrix : \n");

```

```
for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
    {
        printf("%d\t", a[i][j] + b[i][j]);
    }
    printf("\n");
}

// Substraction of Matrix
printf("Substraction  of the matrix : \n");
for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
    {
        printf("%d\t", a[i][j] - b[i][j]);
    }
    printf("\n");
}

// Division of Matrix
printf("Division  of the matrix : \n");
for (i = 0; i < r; i++)
{
    for (j = 0; j < c; j++)
    {
        printf("%d\t", a[i][j] / b[i][j]);
    }
    printf("\n");
}

return 0;
```

```
}
```

Output:-

```
enter the number of row : 3
enter the number of column : 3
enter the first matrix element :
3
2
5
4
6
5
7
8
9
enter the second matrix element :
9
7
6
4
3
2
4
7
8
multiply of the matrix :
55    62    62
80    81    76
131   136   130
Addition of the matrix :
12    9    11
8     9    7
11    15   17
Substraction of the matrix :
-6    -5    -1
0     3     3
3     1     1
Division of the matrix :
0     0     0
1     2     2
1     1     1
```

85 C++\Hamed\Amir_Tajiri\H1\GenPrin\Berkhan\BFA_61m

Practical 8

Code:-

```
// Q 8. WAP to find largest/smallest element of matrix.

#include <stdio.h>

void main()
{
    int mat[10][10];
    int i, j, row, col, small, big;

    printf("Enter the order of the matrix : ");
    scanf("%d %d", &row, &col);
    printf("\nEnter the elements of the matrix : \n\n");
    for (i = 0; i < row; i++)
        for (j = 0; j < col; j++)
            scanf("%d", &mat[i][j]);

    big = mat[0][0];
    small = mat[0][0];
    for (i = 0; i < row; i++)
    {
        for (j = 0; j < col; j++)
        {
            if (mat[i][j] < small)
                small = mat[i][j];
            if (mat[i][j] > big)
                big = mat[i][j];
        }
    }
}
```

```
}

printf("\nThe smallest element in the matrix is : %d\n\n", small);

printf("The Largest element in the matrix is : %d", big);

}
```

Output:-

```
Enter the order of the matrix : 3
3

Enter the elements of the matrix :

2
3
4
5
3
2
4
3
4

The smallest element in the matrix is : 2

The Largest element in the matrix is : 5
```

Practical 9

Code:-

```
// Q 9. WAP to calculate sum of each rows and columns and total
//of all elements of the matrix.
#include <stdio.h>

int main()
{
    int rows, cols, sumRow, sumCol;

    //Initialize matrix a
    int a[][3] = {
        {1, 2, 3},
        {4, 5, 6},
        {7, 8, 9}
    };

    //Calculates number of rows and columns present in given matrix
    rows = (sizeof(a)/sizeof(a[0]));
    cols = (sizeof(a)/sizeof(a[0][0]))/rows;

    //Calculates sum of each row of given matrix
    for(int i = 0; i < rows; i++){
        sumRow = 0;
        for(int j = 0; j < cols; j++){
            sumRow = sumRow + a[i][j];
        }
    }
}
```

```
    }

    printf("Sum of %d row: %d\n", (i+1), sumRow);

}

//Calculates sum of each column of given matrix
for(int i = 0; i < cols; i++){

    sumCol = 0;

    for(int j = 0; j < rows; j++){

        sumCol = sumCol + a[j][i];

    }

    printf("Sum of %d column: %d\n", (i+1), sumCol);

}

return 0;
}
```

Output:-

```
Sum of 1 row: 6
Sum of 2 row: 15
Sum of 3 row: 24
Sum of 1 column: 12
Sum of 2 column: 15
Sum of 3 column: 18
```

Practical 10

Code:-

```
// Q 10. WAP to search element form array using linear search.

#include <stdio.h>

int search(int arr[], int n, int x)
{
    int i;
    for (i = 0; i < n; i++)
        if (arr[i] == x)
            return i;
    return -1;
}

int main(void)
{
    int arr[] = {2, 3, 4, 10, 40};
    int x = 10;
    int n = sizeof(arr) / sizeof(arr[0]);
    // Function call
    int result = search(arr, n, x);
    (result == -1)
        ? printf("Element is not present in array")
        : printf("Element is present at index %d", result);
    return 0;
}
```

Output:- **Element is present at index 3**

Practical 11

Code:-

```
// Q 11. WAP to search element form array using binary search.

#include <stdio.h>

int binarySearch(int arr[], int l, int r, int x)
{
    while (l <= r)
    {
        int m = l + (r - l) / 2;

        // Check if x is present at mid
        if (arr[m] == x)
            return m;

        // If x greater, ignore left half
        if (arr[m] < x)
            l = m + 1;

        // If x is smaller, ignore right half
        else
            r = m - 1;
    }
    return -1;
}
```

```
int main(void)
{
    int arr[] = {2, 3, 4, 10, 40};

    int n = sizeof(arr) / sizeof(arr[0]);

    int x = 10;

    int result = binarySearch(arr, 0, n - 1, x);

    (result == -1) ? printf("Element is not present"
                           " in array")
                  : printf("Element is present at "
                           "index %d",
                           result);

    return 0;
}
```

Output:- **Element is present at index 3**

Practical 12

Code:-

```
// Q 12. WAP to check whether number is palindrome.

#include <stdio.h>

int palindrome(int num)
{
    int temp = 0;
    int duplicate = num;
    while (num > 0)
    {
        int d = num % 10;
        temp = temp * 10 + d;
        num = num / 10;
    }
    // printf("%d",temp);
    if (duplicate == temp)
    {
        return 1;
    }
    else
        return 0;
}

void main()
{
    int num = 0;

    printf("Enter the Number to check Palindrome or Not : ");

    scanf("%d", &num);
```



```
// palindrome(num);  
  
int ans = palindrome(num);  
  
if (ans == 0)  
    printf("Not Palindrome");  
  
else if (ans == 1)  
    printf("Palindrome");  
  
}
```

Output:-

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS C:\Users\Aman Tripathi\OneDrive\Desktop\DSA file> cd "c:\U  
Enter the Number to check Palindrome or Not : 565  
Palindrome  
PS C:\Users\Aman Tripathi\OneDrive\Desktop\DSA file> |
```

Practical 13

Code:-

```
// Q 13. WAP to calculate sum of digit of number.
#include <stdio.h>
int sumOfDigit(int num)
{
    int sum = 0;
    while (num > 0)
    {
        int d = num % 10;
        sum += d;
        num /= 10;
    }
    return sum;
}
int main(int argc, char const *argv[])
{
    int num = 0;
    printf("Enter the Number to find the sum : ");
    scanf("%d", &num);
    int sum = sumOfDigit(num);
    printf("The Sum of Digit of %d is %d.", num, sum);
    return 0;
}
```

Output:-

```
Enter the Number to find the sum : 45654
The Sum of Digit of 45654 is 24.
```

Practical 14

Code:-

```
// Q14. WAP to for Bubble Sort to Sort Elements in An Order.

#include <stdio.h>

void print(int a[], int n) // function to print array elements
{
    int i;
    for (i = 0; i < n; i++)
    {
        printf("%d ", a[i]);
    }
}

void bubble(int a[], int n) // function to implement bubble sort
{
    int i, j, temp;
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (a[j] < a[i])
            {
                temp = a[i];
                a[i] = a[j];
                a[j] = temp;
            }
        }
    }
}
```

```
    }  
}  
void main()  
{  
    int i, j, temp;  
    int a[5] = {10, 35, 32, 13, 26};  
    int n = sizeof(a) / sizeof(a[0]);  
    printf("Before sorting array elements are - \n");  
    print(a, n);  
    bubble(a, n);  
    printf("\nAfter sorting array elements are - \n");  
    print(a, n);  
}
```

Output:-

```
Before sorting array elements are -  
10 35 32 13 26  
After sorting array elements are -  
10 13 26 32 35
```

Practical 15

Code:-

```
// Q15. WAP to for insertion Sort to Sort Elements in An Order.

#include <stdio.h>

void insert(int a[], int n) /* function to sort an aay with insertion
sort */
{
    int i, j, temp;
    for (i = 1; i < n; i++)
    {
        temp = a[i];
        j = i - 1;

        while (j >= 0 && temp <= a[j]) /* Move the elements greater
than temp to one position ahead from their current position*/
        {
            a[j + 1] = a[j];
            j = j - 1;
        }
        a[j + 1] = temp;
    }
}

void printArr(int a[], int n) /* function to print the array */
{
    int i;
    for (i = 0; i < n; i++)
```

```
        printf("%d ", a[i]);  
    }  
  
int main()  
{  
    int a[] = {12, 31, 25, 8, 32, 17};  
    int n = sizeof(a) / sizeof(a[0]);  
    printf("Before sorting array elements are - \n");  
    printArr(a, n);  
    insert(a, n);  
    printf("\nAfter sorting array elements are - \n");  
    printArr(a, n);  
  
    return 0;  
}
```

Output:-

```
Before sorting array elements are -  
12 31 25 8 32 17  
After sorting array elements are -  
8 12 17 25 31 32
```

Practical 16

Code:-

```
// WAP to for selection Sort to Sort Elements in An Order.

#include <stdio.h>

void swap(int arr[], int first, int second)
{
    int temp = arr[first];
    arr[first] = arr[second];
    arr[second] = temp;
}

int getMaxIndex(int arr[], int start, int end)
{
    int max = start;
    for (int i = start; i < end; i++)
    {
        if (arr[max] < arr[i])
        {
            max = i;
        }
    }
    return max;
}

void selectionSort(int arr[], int length)
{
    for (int i = 0; i < length; i++)
    {
        // find the max item in remaining array and swap with correct
        with correct index
        int last = length - i - 1;
        int maxIndex = getMaxIndex(arr, 0, last);
```

```

        swap(arr, maxIndex, last);
    }
}

void main()
{
    int size = 0;
    printf("Enter the Size of array : ");
    scanf("%d", &size);
    int array[size];
    int length = sizeof(array) / sizeof(int);
    printf("Enter the Elements in array ↴");
    printf("\n");
    for (int i = 0; i < size; i++)
    {
        scanf("%d", &array[i]);
    }
    selectionSort(array, size);
    printf("Sorted Elements are ↴ ");
    printf("\n");
    for (int i = 0; i < size; i++)
    {
        printf("%d\t", array[i]);
    }
}

```

Output:-

```

Enter the Size of array : 4
Enter the Elements in array ↴
4
5
6
8
Sorted Elements are ↴
8      4      5      6

```


Practical 17

Code:-

```
// Q17. WAP to for merge Sort to Sort Elements in An Order.

#include <stdio.h>

#include <stdlib.h>

// Merges two subarrays of arr[].
// First subarray is arr[l..m]
// Second subarray is arr[m+1..r]
void merge(int arr[], int l, int m, int r)
{
    int i, j, k;

    int n1 = m - l + 1;

    int n2 = r - m;

    /* create temp arrays */
    int L[n1], R[n2];

    /* Copy data to temp arrays L[] and R[] */
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];

    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    /* Merge the temp arrays back into arr[l..r]*/
    i = 0; // Initial index of first subarray
```

```
j = 0; // Initial index of second subarray
k = 1; // Initial index of merged subarray

while (i < n1 && j < n2) {

    if (L[i] <= R[j]) {

        arr[k] = L[i];

        i++;

    }

    else {

        arr[k] = R[j];

        j++;

    }

    k++;

}

/* Copy the remaining elements of L[], if there
are any */
while (i < n1) {

    arr[k] = L[i];

    i++;

    k++;

}

/* Copy the remaining elements of R[], if there
are any */
while (j < n2) {

    arr[k] = R[j];

    j++;

    k++;

}
```

```

}

/* l is for left index and r is right index of the
sub-array of arr to be sorted */
void mergeSort(int arr[], int l, int r)
{
    if (l < r) {
        // Same as (l+r)/2, but avoids overflow for
        // large l and h
        int m = l + (r - l) / 2;

        // Sort first and second halves
        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);

        merge(arr, l, m, r);
    }
}

/* UTILITY FUNCTIONS */
/* Function to print an array */
void printArray(int A[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ", A[i]);
    printf("\n");
}

```

```
/* Driver code */  
  
int main()  
{  
  
    int arr[] = { 12, 11, 13, 5, 6, 7 };  
  
    int arr_size = sizeof(arr) / sizeof(arr[0]);  
  
    printf("Given array is \n");  
    printArray(arr, arr_size);  
  
    mergeSort(arr, 0, arr_size - 1);  
  
    printf("\nSorted array is \n");  
    printArray(arr, arr_size);  
  
    return 0;  
}
```

Output:-

```
Given array is  
12 11 13 5 6 7
```

```
Sorted array is  
5 6 7 11 12 13
```

Practical 18 & practical 19

Code:-

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char ans;
    ans='Y';
    while(ans=='Y' || ans=='y'){
        int rows,cols;
        printf("Enter the number of rows in the matrix: ");
        scanf("%d", &rows);
        printf("Enter the number of columns in the matrix: ");
        scanf("%d", &cols);
        int arr[rows+1][cols+1];
        int flag, flag1;
        flag=0;
        flag1=0;
        printf("Enter the elements of the matrix: \n");
        for(int i=0;i<rows;++i)
        {
            for(int j=0;j<cols;++j)
            {
                scanf("%d", &arr[i][j]);
            }
        }
        for(int i=0;i<rows;++i)
        {
            for(int j=0;j<cols;++j)
            {
                printf("%d ", arr[i][j]);
            }
        }
    }
}
```

```

    }

    printf("\n");
}

for(int i=0;i<rows;++i)
{
    for(int j=0;j<cols;++j)
    {
        if(j>i && arr[i][j]!=0)
        {
            flag=1;
            break;
        }
    }
}

if(flag==0)
{
    printf("The entered Matrix is a Lower Triangular Sparse
Matrix\n");
}
else
{
    printf("The entered Matrix is not a Lower Triangular Sparse
Matrix\n");
}

for(int i=0;i<rows;++i)
{
    for(int j=0;j<cols;++j)
    {
        if(j<i && arr[i][j]!=0)
        {
            flag1=1;
            break;
        }
    }
}

```

```

    }

    if(flag1==0)
    {
        printf("The entered Matrix is a Upper Triangular Sparse
Matrix\n");
    }
    else
    {
        printf("The entered Matrix is not a Upper Triangular Sparse
Matrix\n");
    }

    printf("If you want to run the program again press Y: ");
    scanf(" %c", &ans);
    if(ans!='Y' && ans!='y')
    {
        printf("Exiting the program");
        exit(0);
    }
    printf("\n");
}

return 0;
}

```

Output:-

```

Enter the number of rows in the matrix: 2
Enter the number of columns in the matrix: 2
Enter the elements of the matrix:
0
0
0
0
0 0
0 0
The entered Matrix is a Lower Triangular Sparse Matrix
The entered Matrix is a Upper Triangular Sparse Matrix
If you want to run the program again press Y: y

```