

Ex No. 2

## Implementation of Matrix ADT

### Aim:

To find the trace and normal of a matrix.

### Algorithm:

1. Start.
2. Receive the number of rows and columns as user input
3. If the number of rows equals the number of columns:
  - a. Declare and initialize the size (in bytes) an 2D matrix array with the number of rows and columns, using pointers
  - b. Prompt user to input values to the array, by a loop construct.
    - i. Successive elements in the matrix construct are input using the pointer definition  $*(arr+i*\text{num\_of\_columns}+j)$ , where i is the outer loop variable and j is the inner loop variable.
  - c. Call the appropriate function for the normal of the matrix.
    - i. Declare a loop with i and j as loop variables
      1. Store the sum of square of each element in the matrix construct in a variable.
      2. Return the variable to the function call
  - d. Print the square root of the returned value as the normal of the matrix.
  - e. Call the appropriate function for the trace of the matrix
    - i. Declare a loop with i and j as loop variables
      1. Check if  $i==j$ .
        - a. If the condition is true, add the value to the sum variable. If false, continue the loop.
      2. Return the variable to the function call.
  - f. Print the returned value as the trace of the matrix.
4. End.

Ex No. 2

Code:

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

int array_normal(int *arr,int rows,int columns){
    int sum=0,val=0;
    for(int i=0;i<rows;i++){
        for(int j=0;j<columns;j++){
            val=*(arr+i*columns+j);
            val=val*val;
            sum+=val;
        }
    }
    return sum;
}

int array_trace(int *arr,int rows,int columns){
    int sum=0,val=0;
    for(int i=0;i<rows;i++){
        for(int j=0;j<columns;j++){
            if(i==j)
            {
                val=*(arr+i*columns+j);
                sum+=val;
            }
        }
    }
    return sum;
}

int main()
{
    int rows,columns;
    printf("Enter the number of rows and columns: \n");
    scanf("%i%i",&rows,&columns);
    if(rows==columns){
        int * arr= (int *)malloc(rows*columns*sizeof(int)); //Allocates bytes to the array
        int i,j,temp;
        printf("Enter the elements one by one: \n");
        for(i=0;i<rows;i++){
```

## Ex No. 2

```
        for(j=0;j<columns;j++){
            scanf("%i",&temp);
            *(arr+i*columns+j)=temp;
        }
    }
    printf("\nThe trace of the matrix is: \n");
    int sum1= array_trace(arr,rows,columns);
    printf("%i",sum1);
    printf("\nThe normal of the matrix is: \n");
    int sum= array_normal(arr,rows,columns);
    printf("%f",sqrt(sum));
}
else{
    printf("Invalid matrix entered!");
}
}
```

Sample Input/Output:

```
Enter the number of rows and columns:
3
3
Enter the elements one by one:
2
3
1
5
3
9
5
6
3

The trace of the matrix is:
8
The normal of the matrix is:
14.106736
```