Name: KAUSHAL VASHISTH Roll Number: 18ETCS002147

Laboratory 2

1. Questions

    1. Write a program to read and perform addition and multiplication of two matrices of order m * n, add them and display the resultant matrix using functions.
    2. Write a program to read a string and check for palindrome without using string related function (a string is palindrome if its half is mirror by itself eg: abcdcba).
    3. Write a program to perform binary search. Use recursion.

2. Introduction

3. Algorithm

Q1:

Step1:- start

Step2:- take input of rows and columns of the matirces and then take input of the elements of matrices using for loop.

Step3:- make function addmatrix in which we have to run 2 nested loops one from (I to r1) and other from(1 to c1). Hence add matrices by logic add[i][j]=m1[i][j]+m2[i][j]; and print the new matrix.

Step4:- make function multiplymatrix in which we have to run 3 nested loops one from (I to r1) and other from(1 to c1) also other from(1 to c1). Hence multiply matrices by mult[i][j]=mult[i][j]+m1[i][k]*m2[k][j];and print the new matrix.

Step5:-call the functions addmatrix and multiplymatrix.

Step6:- stop

Q2:

Step1:- start

Step2:- declare 2 strings s1 and s2, int k=0,l=0;

```
Step3:- take input of s1 string using gets(s1);

Step4:- run a while loop until last element of string is 0. Increment l++ for length.

Step5:- run another for loop to put reverse of string s1 into s2.

Step6:- run another for loop to compare each and every element of s1 and s2.

Step7:- if s1=s2 print("string is palindrome");

            else print("string is not palindrome");

Step8:- stop
```

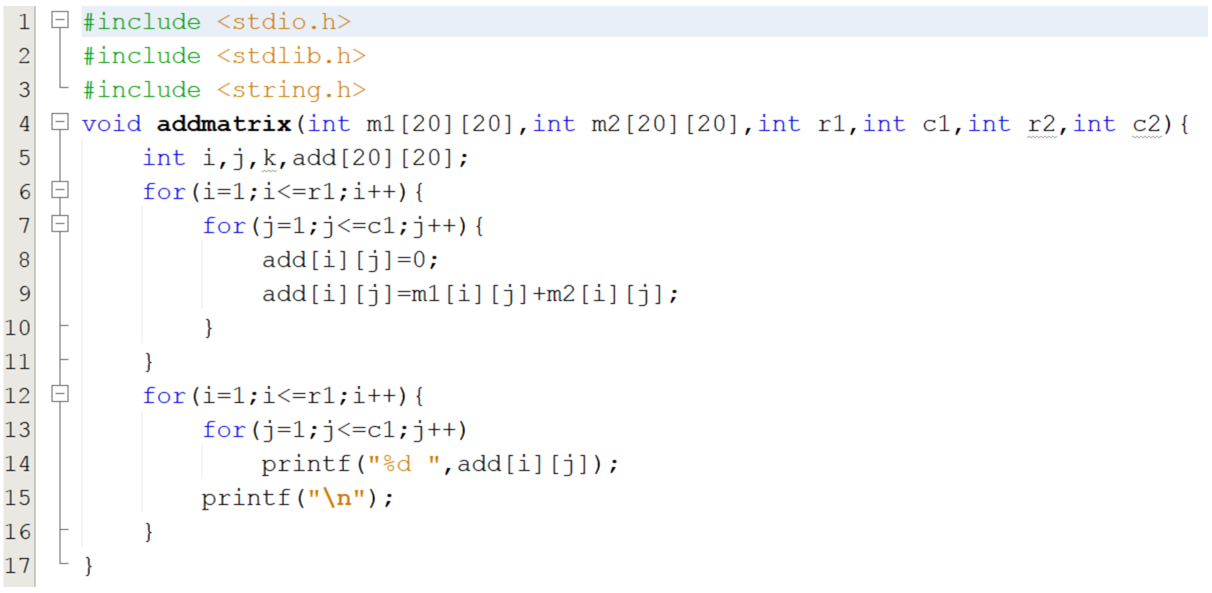
Q3:

```
Step1:- start

Step2:- declare static array {2,3,4,10,40};

Step3:- take x=10 which is the element we want to search.

Step4:- make function binary search() which should follow the following logic:-

    • Compare x with the middle element.
    • If x matches with middle element, we return the mid index.
    • Else If x is greater than the mid element, then x can only lie in right
      half subarray after the mid element. So we recur for right half.
    • Else (x is smaller) recur for the left half.

Step5:- if:- element is present then print its position

            else:- print("element not found");

Step6:- stop
```

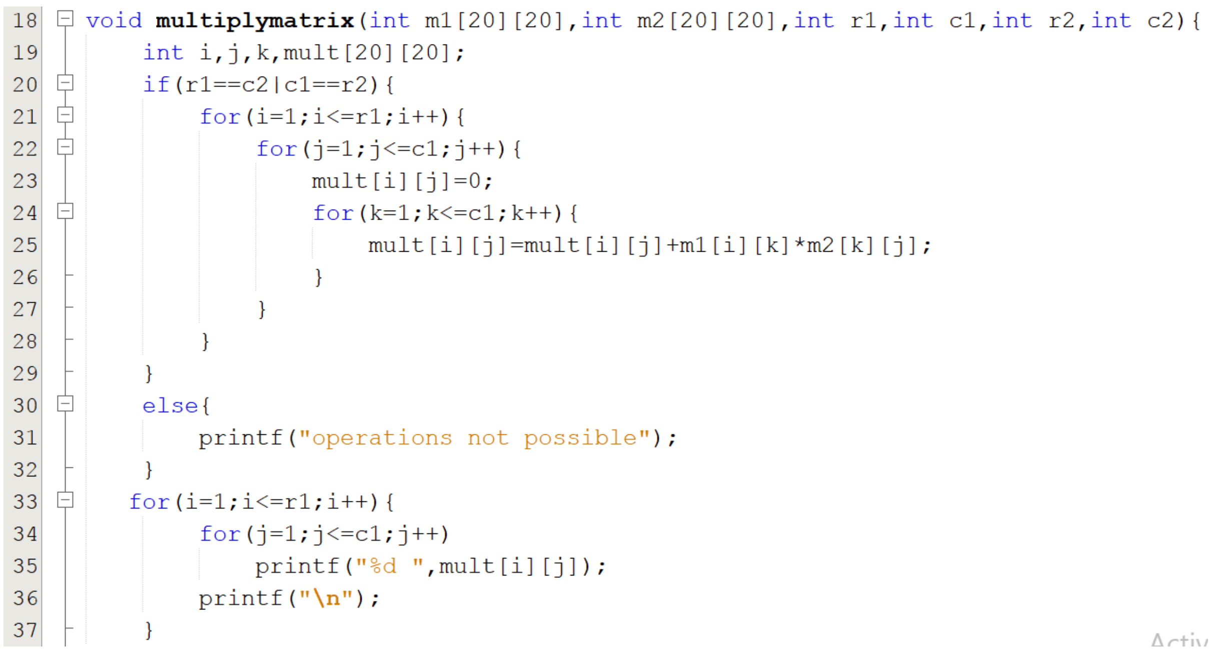
4. Program

Q1:-

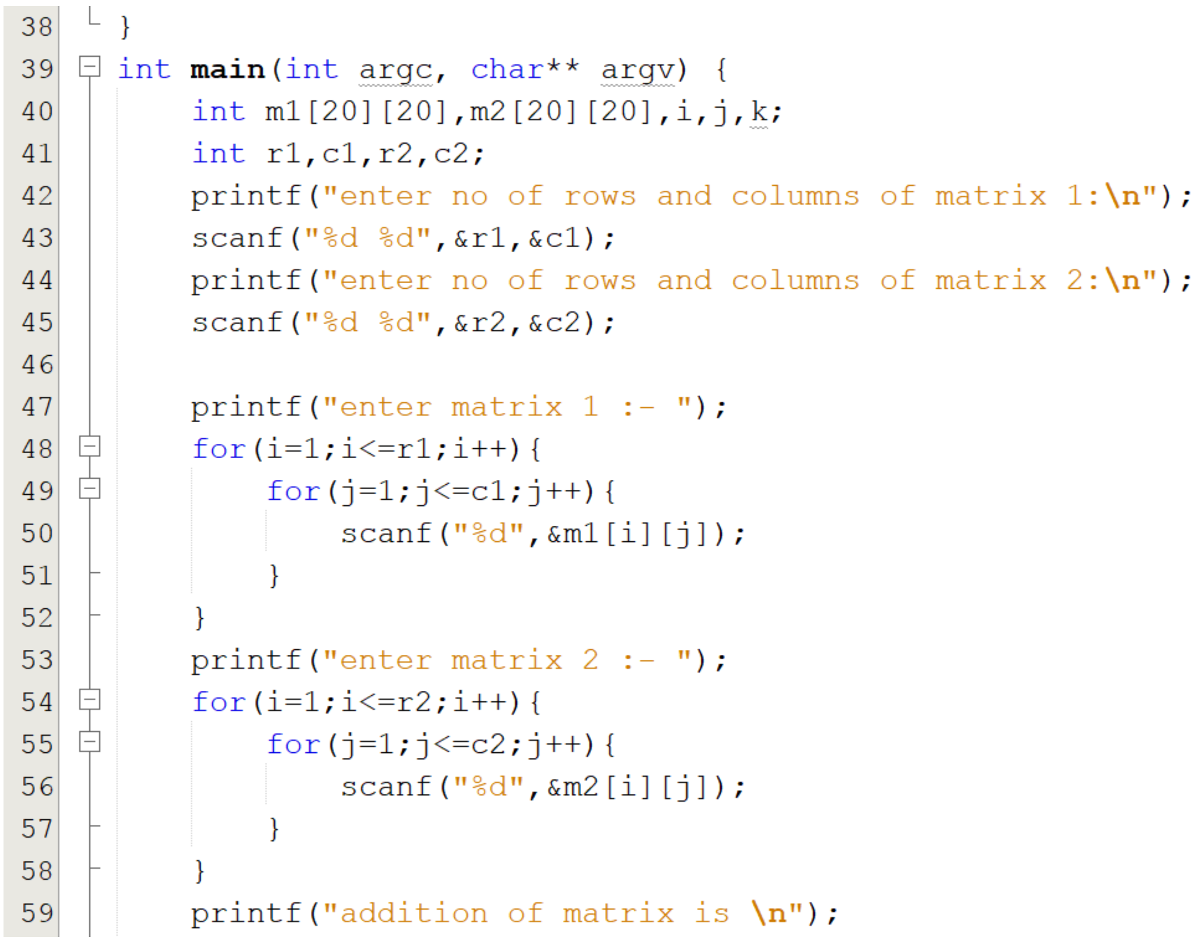
Name: KAUSHAL VASHISTH Roll Number: 18ETCS002147



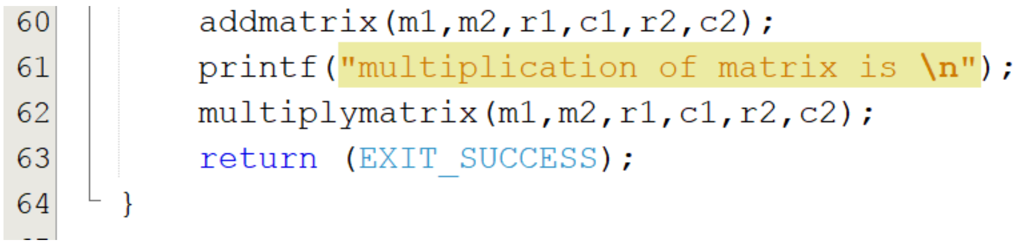
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void addmatrix(int m1[20][20],int m2[20][20],int r1,int c1,int r2,int c2){
    int i,j,k,add[20][20];
    for(i=1;i<=r1;i++){
        for(j=1;j<=c1;j++){
            add[i][j]=0;
            add[i][j]=m1[i][j]+m2[i][j];
        }
    }
    for(i=1;i<=r1;i++){
        for(j=1;j<=c1;j++)
            printf("%d ",add[i][j]);
        printf("\n");
    }
}
```



```c
void multiplymatrix(int m1[20][20],int m2[20][20],int r1,int c1,int r2,int c2){
    int i,j,k,mult[20][20];
    if(r1==c2|c1==r2){
        for(i=1;i<=r1;i++){
            for(j=1;j<=c1;j++){
                mult[i][j]=0;
                for(k=1;k<=c1;k++){
                    mult[i][j]=mult[i][j]+m1[i][k]*m2[k][j];
                }
            }
        }
    }
    else{
        printf("operations not possible");
    }
   for(i=1;i<=r1;i++){
        for(j=1;j<=c1;j++)
            printf("%d ",mult[i][j]);
        printf("\n");
    }
```



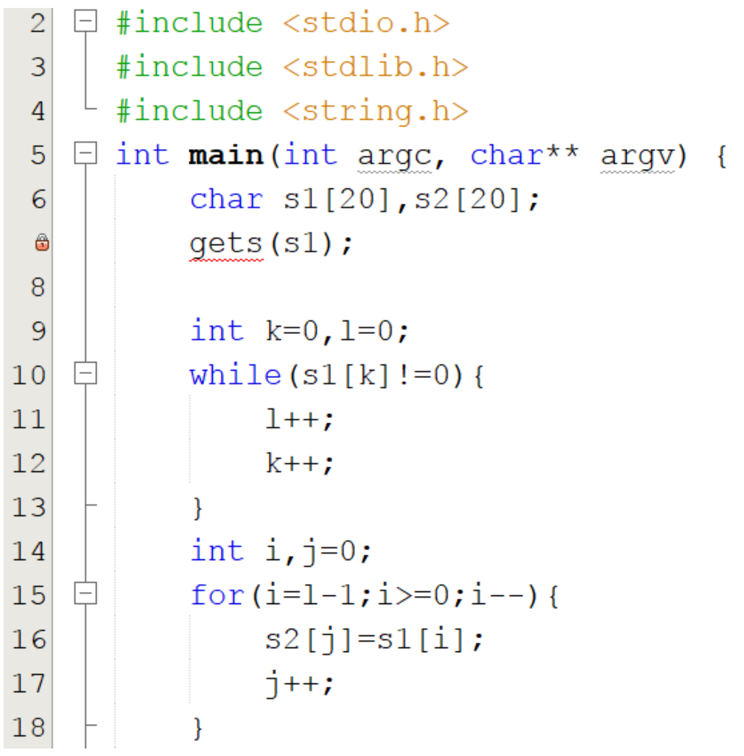
```c
}
int main(int argc, char** argv) {
    int m1[20][20],m2[20][20],i,j,k;
    int r1,c1,r2,c2;
    printf("enter no of rows and columns of matrix 1:\n");
    scanf("%d %d",&r1,&c1);
    printf("enter no of rows and columns of matrix 2:\n");
    scanf("%d %d",&r2,&c2);

    printf("enter matrix 1 :- ");
    for(i=1;i<=r1;i++){
        for(j=1;j<=c1;j++){
            scanf("%d",&m1[i][j]);
        }
    }
    printf("enter matrix 2 :- ");
    for(i=1;i<=r2;i++){
        for(j=1;j<=c2;j++){
            scanf("%d",&m2[i][j]);
        }
    }
    printf("addition of matrix is \n");
```



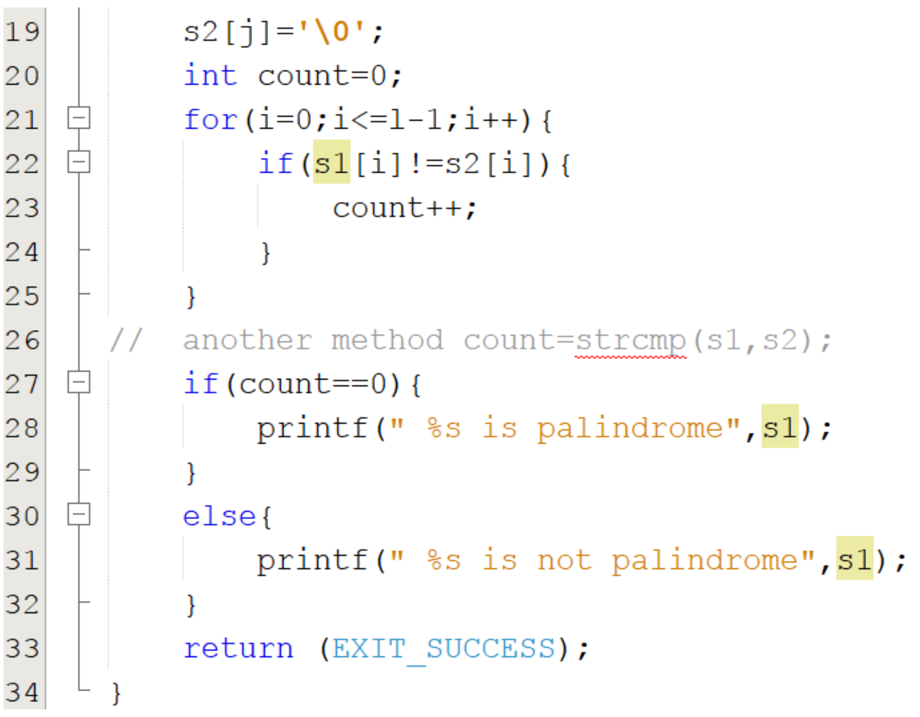
```c
    addmatrix(m1,m2,r1,c1,r2,c2);
    printf("multiplication of matrix is \n");
    multiplymatrix(m1,m2,r1,c1,r2,c2);
    return (EXIT_SUCCESS);
}
```

Q2:-

Name: KAUSHAL VASHISTH   Roll Number: 18ETCS002147

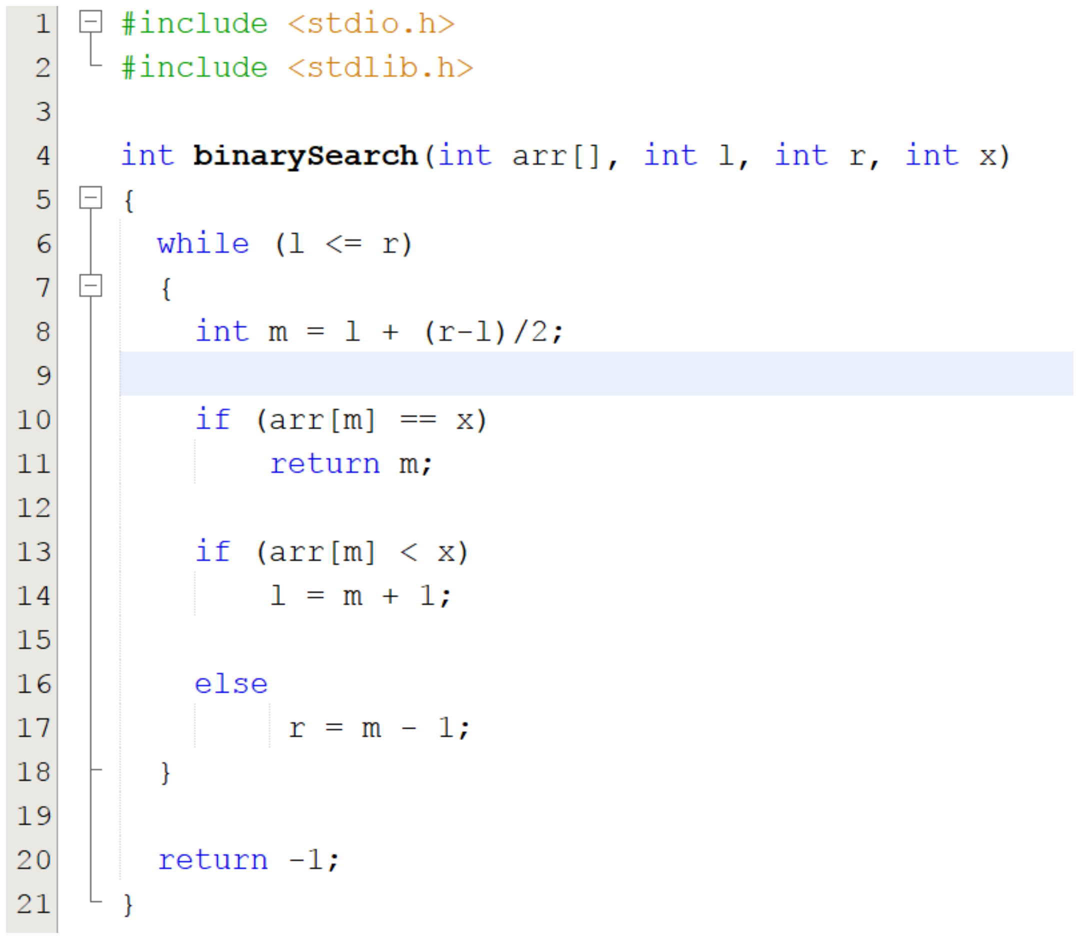


```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(int argc, char** argv) {
    char s1[20],s2[20];
    gets(s1);

    int k=0,l=0;
    while(s1[k]!=0){
        l++;
        k++;
    }
    int i,j=0;
    for(i=l-1;i>=0;i--){
        s2[j]=s1[i];
        j++;
    }
```

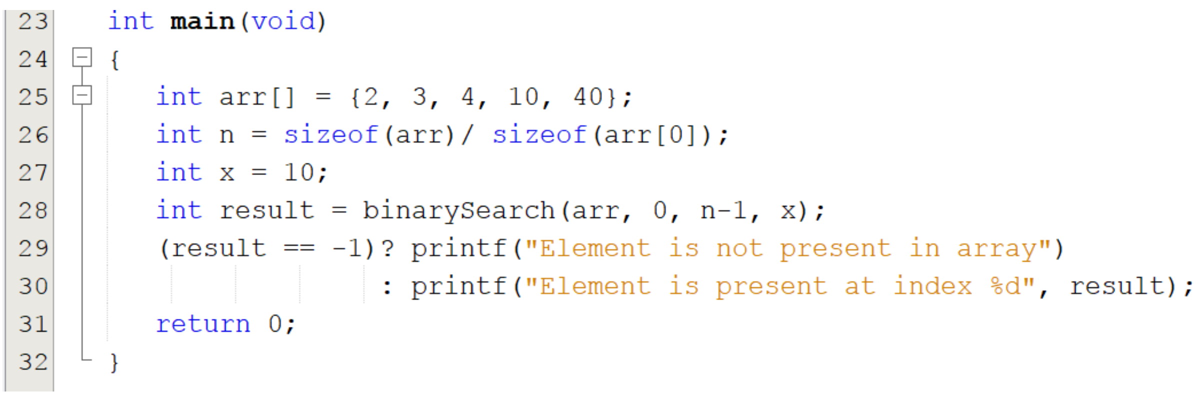


```c
    s2[j]='\0';
    int count=0;
    for(i=0;i<=l-1;i++){
        if(s1[i]!=s2[i]){
            count++;
        }
    }
//  another method count=strcmp(s1,s2);
    if(count==0){
        printf(" %s is palindrome",s1);
    }
    else{
        printf(" %s is not palindrome",s1);
    }
    return (EXIT_SUCCESS);
}
```

Q3:-



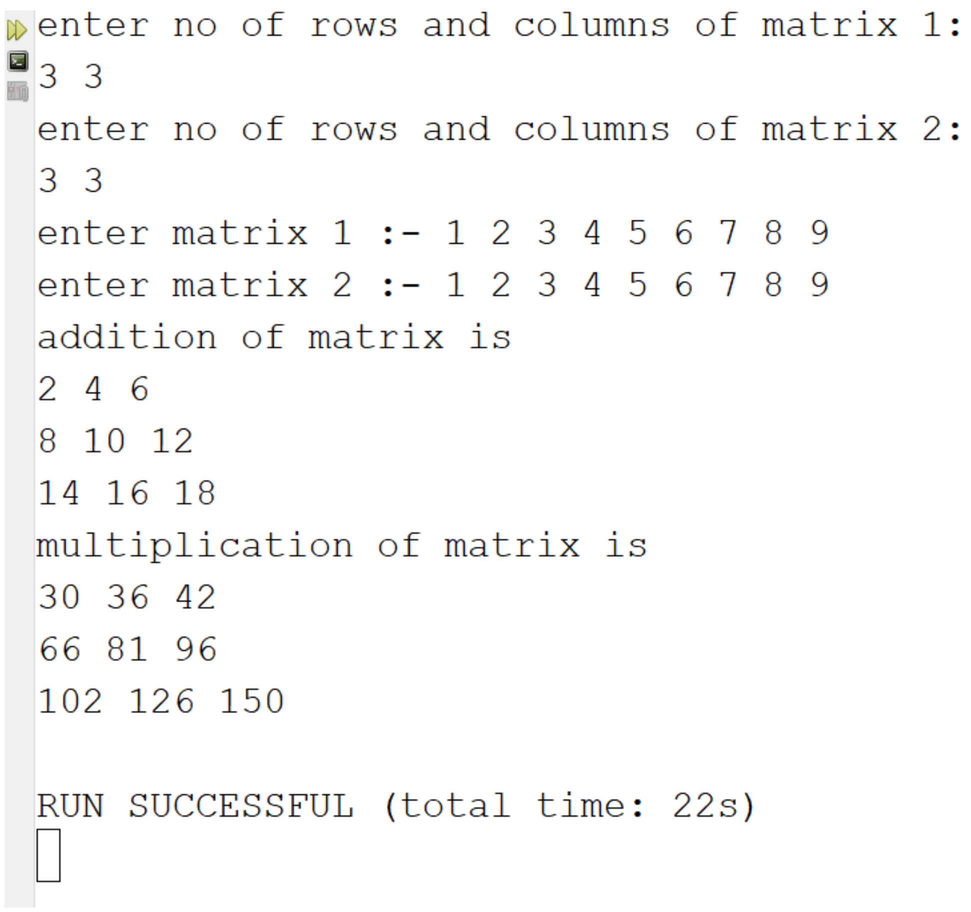
```c
#include <stdio.h>
#include <stdlib.h>

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

      if (arr[m] == x)
          return m;

      if (arr[m] < x)
          l = m + 1;

      else
           r = m - 1;
    }

    return -1;
}
```



```c
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;
}
```

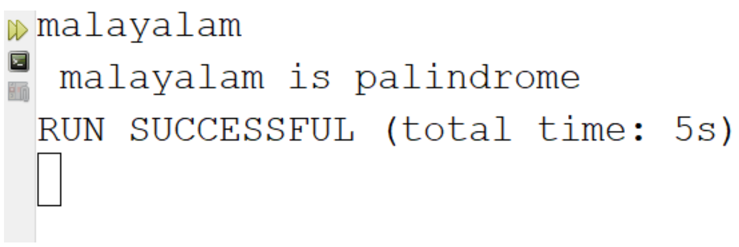
5. Presentation of Results:-

Q1:-

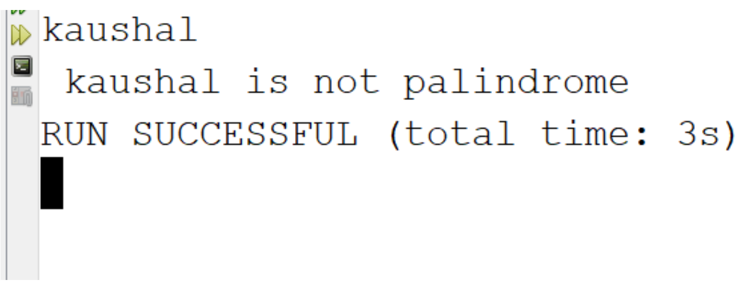


```
enter no of rows and columns of matrix 1:
3 3
enter no of rows and columns of matrix 2:
3 3
enter matrix 1 :- 1 2 3 4 5 6 7 8 9
enter matrix 2 :- 1 2 3 4 5 6 7 8 9
addition of matrix is
2 4 6
8 10 12
14 16 18
multiplication of matrix is
30 36 42
66 81 96
102 126 150

RUN SUCCESSFUL (total time: 22s)
```

Q2:-

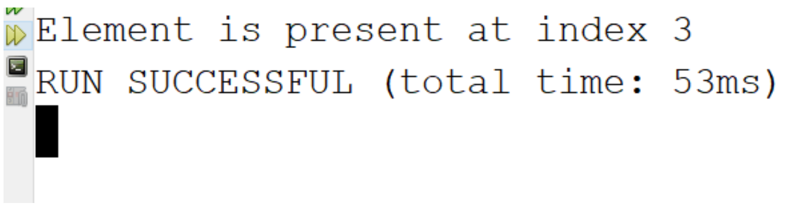


```
malayalam
 malayalam is palindrome
RUN SUCCESSFUL (total time: 5s)
```



```
kaushal
 kaushal is not palindrome
RUN SUCCESSFUL (total time: 3s)
```

Q3:-



```
Element is present at index 3
RUN SUCCESSFUL (total time: 53ms)
```

6. Conclusions :-

We can conclude that all the programs have been executed without any errors. This experiment gave us brief about multiplying and adding matrices, to check a string is palindrome or not and to perform binary search using recursion.