# 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. Algorithm

**2.1 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.**

Step1: start

Step2: input no. of rows and columns for first and second matrix

Step3: input matrices

Step4: add matrix

4.1 for(int i = 0; i < r1; i++)

4.2 for(int j = 0; j < c2; j++)

4.3 add[i][j] = m1[i][j]+m2[i][j];

Step5: multiply matrix

5.1 for(i = 0; i < r1; i++){

5.2 for(j = 0; j < c2; j++){

5.3 for(k=0; k<c1; k++){

5.4 mult[i][j] += m1[i][k] \* m2[k][j];

Step6: print added and multiplied matrix

6.1 for(int i = 0; i < m; i++)

6.2 {for(int j = 0; j < n; j++)

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

}printf("\n");}}

Step7: stop

**2.2 a program to read a string and check for palindrome without using string related function**

Step1: start

Step2: input string

Step3: calculate length of the string

3.1 for ( i=0;st[i]!='\0';i++)

{len++;}

Step4: find reverse of the string

4.1 for (i=len-1; i >= 0; i--)

{rev[len-i-1]=st[i]; }

Step5: check if both the reversed and the original matrix is same or not

Step6: print if palindrome or not

Step7: stop

**2.3 a program to perform binary search. Use recursion.**

Step1: start

Step2: input size of array

Step3: input elements of the arrays

Step4: input target element

Step5: build a function

int BSR(int a[],int low,int high,int tar){

int mid = (low+high)/2;

Step6: if (low>high || high<low){

printf("Not found!!");

return 0;

}

Step7: if (tar > a[mid]) {

low = mid+1;

BSR(a,low,high,tar);

}

Step8: if (tar < a[mid]) {

high = mid-1;

BSR(a,low,high,tar);

}

Step9: if (tar == a[mid]) {

printf("%d was found at position %d\n",tar,mid );

return 1;

}

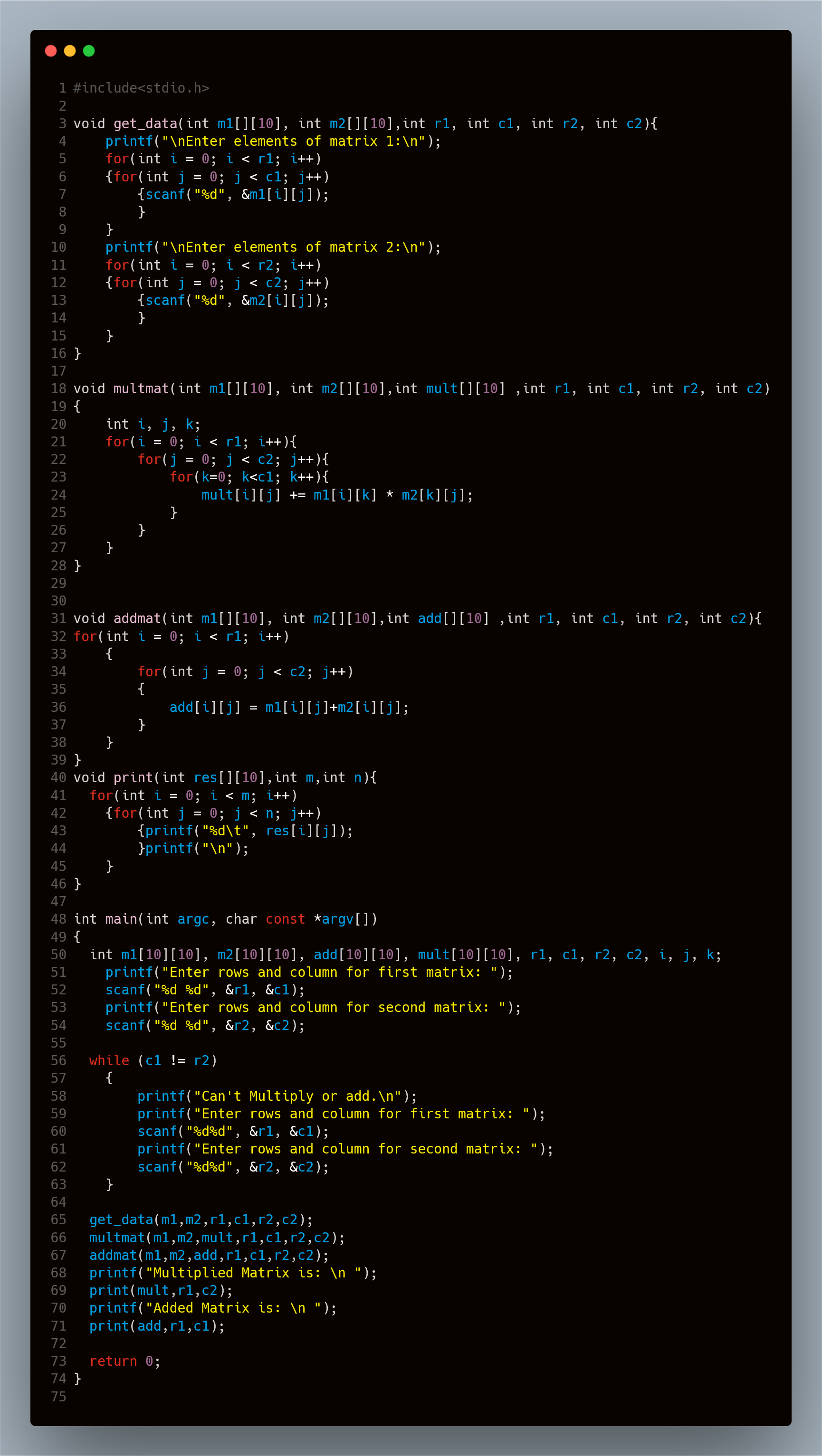
Step10: call function

10.1 BSR(a,0,(n-1),tar);

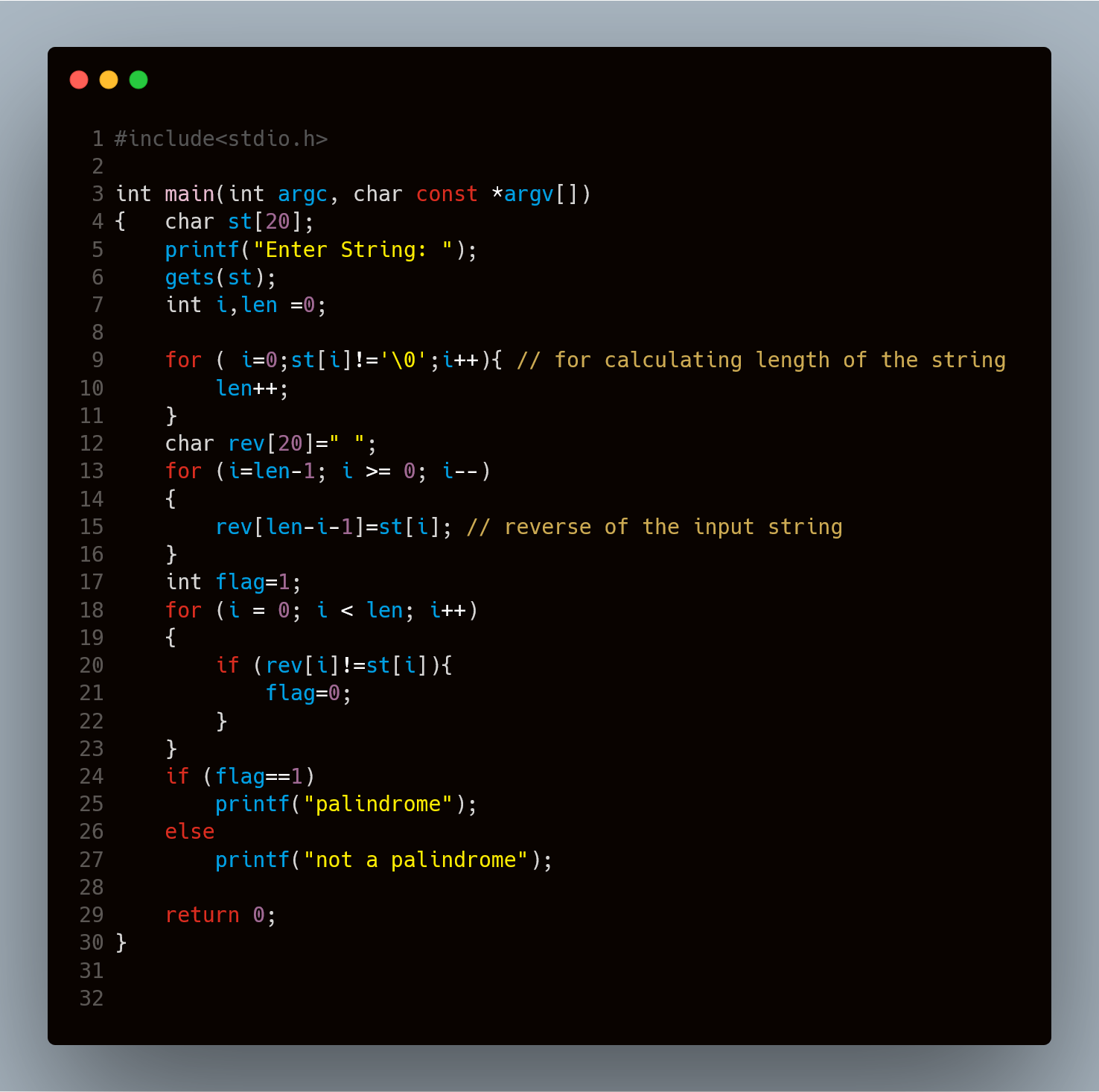
Step11: Stop

1. Program

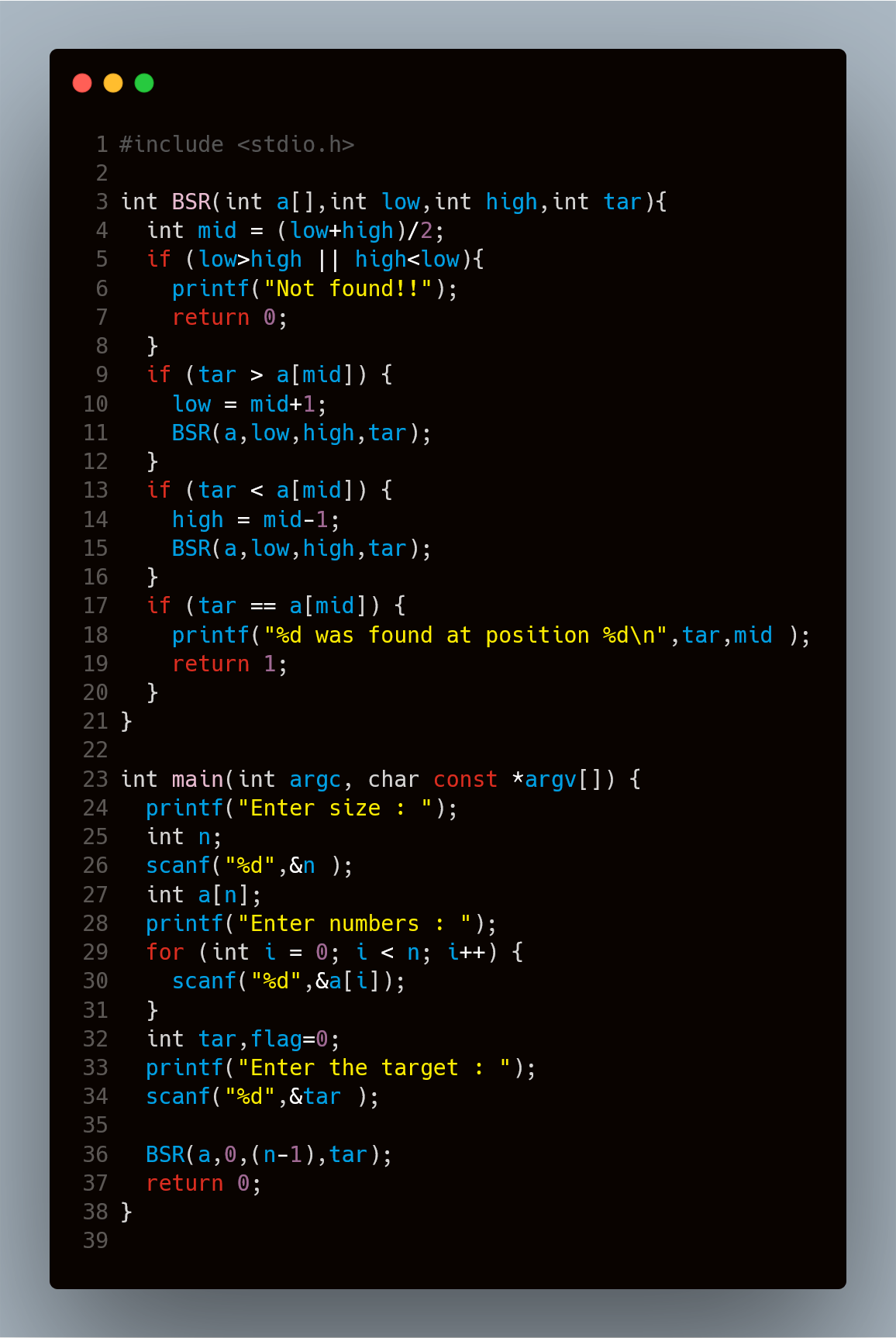
3.1 Program to read and perform addition and multiplication of two matrices of order m \* n, add them and display the resultant matrix using functions.



3.2 Program to read a string and check for palindrome without using string related function

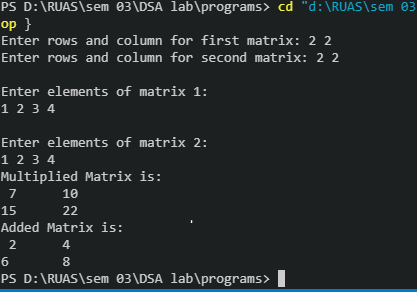


3.3 Program to perform binary search using recursion.

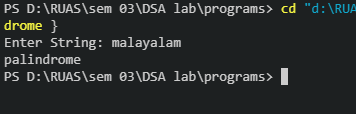


1. Presentation of Results

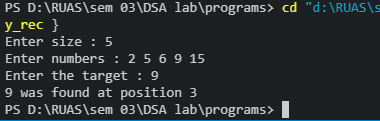
4.1 Program to read and perform addition and multiplication of two matrices of order m \* n, add them and display the resultant matrix using functions.



4.2 Program to read a string and check for palindrome without using string related function



4.3 Program to perform binary search using recursion.



1. Conclusions

Hence we can see the programs are compiled successfully without any error.