Name: SUBHENDU MAJI Roll Number: 18ETCS002121

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
```

3. 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.

Name: SUBHENDU MAJI Roll Number: 18ETCS002121

```c
#include<stdio.h>

void get_data(int m1[][10], int m2[][10],int r1, int c1, int r2, int c2){
    printf("\nEnter elements of matrix 1:\n");
    for(int i = 0; i < r1; i++)
    {for(int j = 0; j < c1; j++)
        {scanf("%d", &m1[i][j]);
        }
    }
    printf("\nEnter elements of matrix 2:\n");
    for(int i = 0; i < r2; i++)
    {for(int j = 0; j < c2; j++)
        {scanf("%d", &m2[i][j]);
        }
    }
}

void multmat(int m1[][10], int m2[][10],int mult[][10] ,int r1, int c1, int r2, int c2)
{
    int i, j, k;
    for(i = 0; i < r1; i++){
        for(j = 0; j < c2; j++){
            for(k=0; k<c1; k++){
                mult[i][j] += m1[i][k] * m2[k][j];
            }
        }
    }
}


void addmat(int m1[][10], int m2[][10],int add[][10] ,int r1, int c1, int r2, int c2){
for(int i = 0; i < r1; i++)
    {
        for(int j = 0; j < c2; j++)
        {
            add[i][j] = m1[i][j]+m2[i][j];
        }
    }
}
void print(int res[][10],int m,int n){
  for(int i = 0; i < m; i++)
    {for(int j = 0; j < n; j++)
        {printf("%d\t", res[i][j]);
        }printf("\n");
    }
}

int main(int argc, char const *argv[])
{
  int m1[10][10], m2[10][10], add[10][10], mult[10][10], r1, c1, r2, c2, i, j, k;
    printf("Enter rows and column for first matrix: ");
    scanf("%d %d", &r1, &c1);
    printf("Enter rows and column for second matrix: ");
    scanf("%d %d", &r2, &c2);

  while (c1 != r2)
    {
        printf("Can't Multiply or add.\n");
        printf("Enter rows and column for first matrix: ");
        scanf("%d%d", &r1, &c1);
        printf("Enter rows and column for second matrix: ");
        scanf("%d%d", &r2, &c2);
    }

  get_data(m1,m2,r1,c1,r2,c2);
  multmat(m1,m2,mult,r1,c1,r2,c2);
  addmat(m1,m2,add,r1,c1,r2,c2);
  printf("Multiplied Matrix is: \n ");
  print(mult,r1,c2);
  printf("Added Matrix is: \n ");
  print(add,r1,c1);

  return 0;
}

```

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

```c
#include<stdio.h>

int main(int argc, char const *argv[])
{   char st[20];
    printf("Enter String: ");
    gets(st);
    int i,len =0;

    for ( i=0;st[i]!='\0';i++){ // for calculating length of the string
        len++;
    }
    char rev[20]=" ";
    for (i=len-1; i >= 0; i--)
    {
        rev[len-i-1]=st[i]; // reverse of the input string
    }
    int flag=1;
    for (i = 0; i < len; i++)
    {
        if (rev[i]!=st[i]){
            flag=0;
        }
    }
    if (flag==1)
        printf("palindrome");
    else
        printf("not a palindrome");

    return 0;
}
```

3.3 Program to perform binary search using recursion.

```c
#include <stdio.h>

int BSR(int a[],int low,int high,int tar){
  int mid = (low+high)/2;
  if (low>high || high<low){
    printf("Not found!!");
    return 0;
  }
  if (tar > a[mid]) {
    low = mid+1;
    BSR(a,low,high,tar);
  }
  if (tar < a[mid]) {
    high = mid-1;
    BSR(a,low,high,tar);
  }
  if (tar == a[mid]) {
    printf("%d was found at position %d\n",tar,mid );
    return 1;
  }
}

int main(int argc, char const *argv[]) {
  printf("Enter size : ");
  int n;
  scanf("%d",&n );
  int a[n];
  printf("Enter numbers : ");
  for (int i = 0; i < n; i++) {
    scanf("%d",&a[i]);
  }
  int tar,flag=0;
  printf("Enter the target : ");
  scanf("%d",&tar );

  BSR(a,0,(n-1),tar);
  return 0;
}

```

4. 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.

```
PS D:\RUAS\sem 03\DSA lab\programs> cd "d:\RUAS\sem 03
op }
Enter rows and column for first matrix: 2 2
Enter rows and column for second matrix: 2 2

Enter elements of matrix 1:
1 2 3 4

Enter elements of matrix 2:
1 2 3 4
Multiplied Matrix is:
 7      10
15      22
Added Matrix is:
 2      4
6       8
PS D:\RUAS\sem 03\DSA lab\programs>
```

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

```
PS D:\RUAS\sem 03\DSA lab\programs> cd "d:\RUA
drome }
Enter String: malayalam
palindrome
PS D:\RUAS\sem 03\DSA lab\programs>
```

4.3 Program to perform binary search using recursion.

```
PS D:\RUAS\sem 03\DSA lab\programs> cd "d:\RUAS\s
y_rec }
Enter size : 5
Enter numbers : 2 5 6 9 15
Enter the target : 9
9 was found at position 3
PS D:\RUAS\sem 03\DSA lab\programs>
```

5. Conclusions

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