Name: DEEPAK R Roll Number:18ETCS002041

Laboratory 1

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

Binary search explanation

A Binary Search is a quick and efficient method of finding a specific target value from a set of ordered items. A Binary Search, also known as a half-interval search. 2. In Binary Search the key value which is to be searched is compared to the middle element of the array. If the key value is less than or greater than this middle element, the algorithm knows which half of the array to continue searching in because the array is sorted. 3. This process is repeated until we discover the element. In each step this algorithm divides the array size by half and the Binary search will be successful if it is able to locate the element in array, but if it cannot find the element in array it simply returns -1 or prints "Key notFound".

4. Worst case time complexity of Binary Search it is O(log n). The Best case time complexity of Binary Search is O(1). The only condition for implementing Binary Search is that the array should be sorted.

Palindrome explanation

A string is said to be palindrome if reverse of the string is same as string. For example, "abba" is palindrome, but "abbc" is not palindrome.

Matrix problem explanation

Matrix addition is the operation of adding two matrices by adding the corresponding entries together. The matrix can be added only when the number of rows and columns of the first matrix is equal to the number of rows and columns of the second matrix.

We can multiply two matrices if, and only if, the number of columns in the first matrix equals the number of rows in the second matrix. Otherwise, the product of two matrices is undefined.

Name: DEEPAK R Roll Number:18ETCS002041

3. Algorithm

1.Algorithm for binary search using recursion

1. Find the midpoint of the array; this will be the element at *arr[size/2].* The midpoint divides the array into two smaller arrays: the lower half of the array consisting of elements 0 to midpoint - 1, and the upper half of the array consisting of elements midpoint to size - 1.
2. Compare *key* to *arr[midpoint]* by calling the user function *cmp_proc*.
3. If the key is a match, return *arr[midpoint];* otherwise
4. If the array consists of only one element return NULL, indicating that there is no match; otherwise
5. If the key is less than the value extracted from *arr[midpoint]* search the lower half of the array by recursively calling *search;* otherwise
6. Search the upper half of the array by recursively calling *search*

2.Algorithm to check whether given string is palindrome or not

- Input the string.
- Find the reverse of the string.
- If the reverse of the string is equal to the input string, then return true. Else, return false.

4. Algorithm for matrix multiplication

1. Start
2. Declare variables and initialize necessary variables
3. Enter the element of matrices by row wise using loops
4. Check the number of rows and column of first and second matrices
5. If number of rows of first matrix is equal to the number of columns of second matrix, go to step 6. Otherwise, print matrix multiplication is not possible and go to step 3.
6. Multiply the matrices using nested loops.
7. Print the product in matrix form as console output.
8. Stop

5. Algorithm for matrix addition
    1. We first define three matrices A, B, C and read their respective row and column numbers in variable r and c
    2. Read matrices A and B.
    3. First, start a loop for getting row elements of A and B
       Secondly, inside it again start a loop for column of A and B
    4. Then, we store their corresponding addition by C[i][j]=A[i][j] + B[i][j] into C[i][j]
6. Program

```c
/**program done by Deepak R 18ETCs002041*/
#include <stdio.h>
#include<string.h>
#include <stdlib.h>
int main(int argc, char** argv) {
char text[100];
    int begin,middle,end,length=0;
    printf("enter any string to check its pallindrome \n");
    gets(text);
    while(text[length] != '\0')
        length++ ;
    end=length-1;
    middle=length/2;
            for (begin=0; begin<middle; begin++)
            {
                if(text[begin]!=text[end])
                {
                    printf("given string is not a pallindrome \n");
                break;

                }
                    end--;
                }
    if (begin==middle)
        printf(" given string is pallindrome. \n");
    return (EXIT_SUCCESS);
}
```

Fig 1 program to check whether given string is palindrome or not

```c
/*program done by Deepak R 18ETCS002041*/
#include <stdio.h>
#include <stdlib.h>
#define size 10
int binsearch(int[], int, int, int);
int main(){
    int num, i, key, position;
    int low, high, list[size];
    printf("\nEnter the total number of elements");
    scanf("%d",&num);
    printf("\nEnter the elements of list :");
    for(i=0;i<num;i++){
        scanf("%d",&list[i]);
    }
    low = 0;
    high = num-1;
    printf("\nEnter the elements to be searched :");
    scanf("%d",&key);
    position = binsearch(list,key,low,high);
    if(position!=-1){
        printf("\nNumber present at %d",(position+1));
    }else
        printf("\nThe number is not present in the list");
    return(0);
}
int binsearch(int a[], int x, int low, int high){
    int mid;
    if(low>high)
        return -1;
    mid=(low+high)/2;
    if(x==a[mid]){
        return(mid);
    }else if(x<a[mid]){
        binsearch(a,x,low,mid-1);
    }else{
        binsearch(a,x,mid+1,high);
    }
}
```

Fig 2 program for Binary search

```c
#include <stdio.h> /*program done by Deepak R */
int main()
{
   int m, n, c, d, first[10][10], second[10][10], sum[10][10];

   printf("Enter the number of rows and columns of matrix\n");
   scanf("%d%d", &m, &n);
   printf("Enter the elements of first matrix\n");

   for (c = 0; c < m; c++)
      for (d = 0; d < n; d++)
         scanf("%d", &first[c][d]);

   printf("Enter the elements of second matrix\n");

   for (c = 0; c < m; c++)
      for (d = 0 ; d < n; d++)
         scanf("%d", &second[c][d]);

   printf("Sum of entered matrices:-\n");

   for (c = 0; c < m; c++) {
      for (d = 0 ; d < n; d++) {
         sum[c][d] = first[c][d] + second[c][d];
         printf("%d\t", sum[c][d]);
      }
      printf("\n");
   }
   return 0;
}
```

Fig 3 Program for matrix Addition

```c
#include <stdio.h>
int main()
{
  int m, n, p, q, c, d, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];

  printf("Enter number of rows and columns of first matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter elements of first matrix\n");

  for (c = 0; c < m; c++)
    for (d = 0; d < n; d++)
      scanf("%d", &first[c][d]);

  printf("Enter number of rows and columns of second matrix\n");
  scanf("%d%d", &p, &q);

  if (n != p)
    printf("The matrices can't be multiplied with each other.\n");
  else
  {
    printf("Enter elements of second matrix\n");

    for (c = 0; c < p; c++)
      for (d = 0; d < q; d++)
        scanf("%d", &second[c][d]);

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++) {
        for (k = 0; k < p; k++) {
          sum = sum + first[c][k]*second[k][d];
        }
        multiply[c][d] = sum;
        sum = 0;
      }
    }
    printf("Product of the matrices:\n");

    for (c = 0; c < m; c++) {
      for (d = 0; d < q; d++)
        printf("%d\t", multiply[c][d]);

      printf("\n");
    }
  }
 return 0;
}
```

Fig 4 program for matrix multiplication

7. Presentation of Results

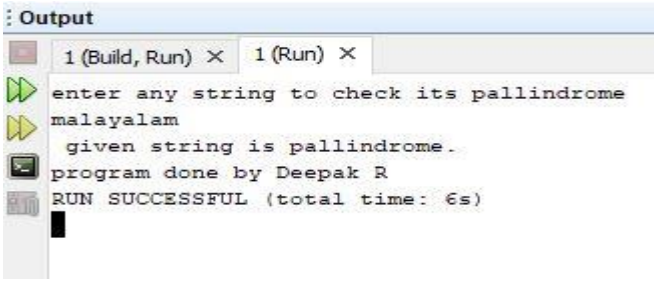

Fig 5 result of program to check whether given string is palindrome or not

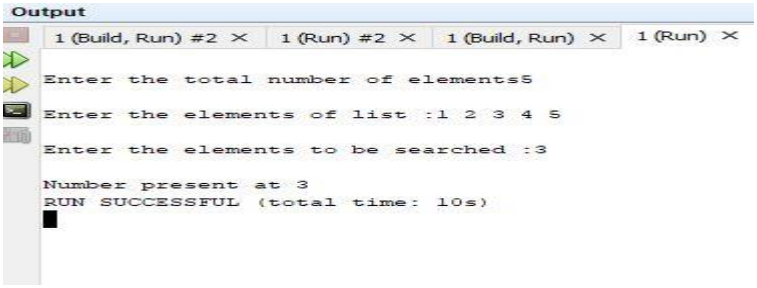

Fig 6 Result of program for Binary search

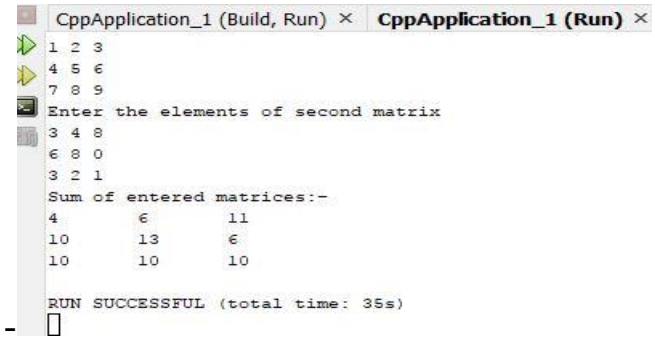

Fig 7 Result of matrix addition

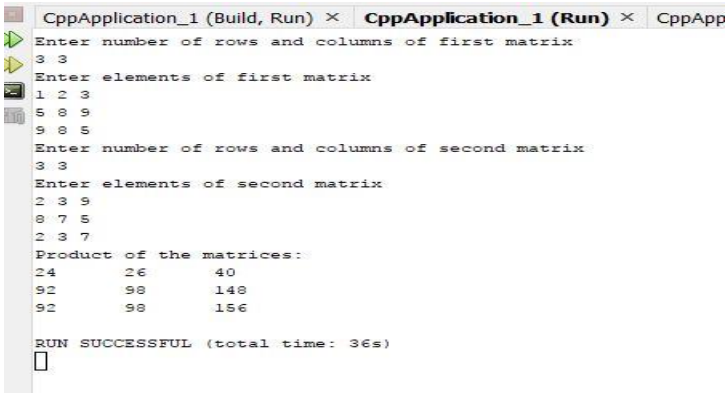

Fig 8 Result of matrix Multiplication

8. Conclusions

This is the first code written in C program. The program is focused on Binary search and program to input two matrix and display the sum and multiplication of between these two matrices .and a program to check whether given string is palindrome or not.From this lab, I understood the basic structure of C programming including the meaning of header files & steps of problem solving.