1BM19CS176

Binary search

```
#include<stdio.h>
void binary_search(int[],int,int,int);
void bubble_sort(int[],int);
int main()
{
   int key,size,i;
   int list[25];
   printf("Enter size of a list:");
   scanf("%d",&size);
   printf("Enter elements\n");
   for(i=0;i<size;i++)
   {
      scanf("%d",&list[i]);
   }
   bubble_sort(list,size);
   printf("\n");
   printf("Enter key to search\n");
   scanf("%d",&key);
   binary_search(list,0,size,key);
}
void bubble_sort(int list[],int size)
{
   int temp,i,j;
   for(i=0;i<size;i++)
   {
      for(j=i;j<size;j++)
      {
         if(list[i]>list[j])
         {
            temp=list[i];
            list[i]=list[j];
            list[j]=temp;
         }
      }
   }
}
void binary_search(int list[],int lo,int hi,int key)
{
   int mid;
   if(lo>hi)
   {
      printf("Key not found\n");
```

```
        return;
    }
    mid=(lo+hi)/2;
    if(list[mid]==key)
    {
        printf("Key found\n");
    }
    else if(list[mid]>key)
    {
        binary_search(list,lo,mid-1,key);
    }
    else if(list[mid]<key)
    {
        binary_search(list,mid+1,hi,key);
    }
}
```

```
Enter size of a list:4
Enter elements
12
78
903
56

Enter key to search
78
Key found
```

```
Enter size of a list:6
Enter elements
121
54
33
28
97
107

Enter key to search
67
Key not found
```

GCD

```
#include <stdio.h>
int gcd(int m,int n)
```

```
{
    int r;
    do{
      r=m%n;
      m=n;
      n=r;
    } while(n!=0);
    return m;
}
int main()
{
    int m,n,res;
    printf("Enter m and n\n");
    scanf("%d%d",&m,&n);
    res=gcd(m,n);
    printf("The GCD of %d and %d is %d.\n",m,n,res);

}
```

GCD

```
#include <stdio.h>
int gcd(int m,int n)
{
    if(n==0) return m;
    if(m<n) return gcd(n,m);
    return gcd(n,m%n);
}
int main()
{
    int m,n,res;
    printf("Enter m and n\n");
    scanf("%d%d",&m,&n);
    res=gcd(m,n);
    printf("GCD of %d and %d is %d.\n",m,n,res);
}
```

TOWER OF HANOI

```
#include<time.h>
#include <stdio.h>
 int TOH(int,char,char,char);
 int main()
 {
   int n;
        clock_t t;
    t = clock();
   printf("\nEnter number of plates:");
   scanf("%d",&n);
    int c = TOH(n,'A','C','B');
    printf("\n");
    printf("Total number of moves = %d \n ", c);
     t = clock() - t;
    double time_taken = ((double)t)/CLOCKS_PER_SEC; // in seconds
    printf("ALGO took %f seconds to execute \n", time_taken);
  return 0;
  }
  int TOH(int n,char first,char third,char second)
 {
   int count;
  if(n>0){
     count=TOH(n-1, first, second, third);
     printf("Move disk %d from peg %c to peg %c\n", n, first, third);
     count++;
     count+= TOH(n-1, second, third, first);
     }
  return count;
 }
```

LINEAR SEARCH

```
#include <stdio.h>
int RecursiveLS(int arr[], int value, int index, int n)
```

```
{
    int pos = 0;

    if(index >= n)
    {
        return 0;
    }

    else if (arr[index] == value)
    {
        pos = index + 1;
        return pos;
    }

    else
    {
        return RecursiveLS(arr, value, index+1, n);
    }
    return pos;
}

int main()
{
    int n, value, pos, m = 0, arr[100];
    printf("Enter the total elements in the array: ");
    scanf("%d", &n);

    printf("Enter the array elements: \n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }

    printf("Enter the element to search:  ");
    scanf("%d", &value);

    pos =  RecursiveLS(arr, value, 0, n);
    if (pos != 0)
    {
        printf("Element found at pos %d\n", pos);
    }
    else
    {
        printf("Element not found\n");
```

```
    }
    return 0;
}
```

```
Enter the total elements in the array: 7
Enter the array elements:
23
908
76
12
6542
89
0
Enter the element to search:  908
Element found at pos 2
```

```
Enter the total elements in the array: 3
Enter the array elements:
89
765
123
Enter the element to search:  76
Element not found
```