**PROGRAM-2  
Recursive Binary search and Linear search and determine the time required to  
search an element. Repeat the experiment for different values of N  and plot a graph of the time taken versus N.   
Last class you have done the 2nd program, for the same program determine the time required by incorporating  functions from time.h**

#include<stdio.h>

#include<stdlib.h>

#include<time.h>

int linear(int ele, int num, int a[], int index){

if(index>=num)

return 0;

if(ele==a[index])

return (index+1);

else

return linear(ele,num,a,index+1);

}

void sort(int num, int a[]){ // descending order //

int j,k,temp;

for(j=0;j<num-1;j++)

for(k=j+1;k<num;k++){

if(a[k]>a[j]){

temp=a[j];

a[j]=a[k];

a[k]=temp;

}

}

}

int binary(int ele, int a[],int low, int high){

int mid;

mid=(low+high)/2;

if (low>high)

return 0;

if (ele==a[mid])

return (mid+1);

else if(a[mid]>ele)

return binary(ele,a,mid+1,high);

else

return binary(ele,a,low,mid-1);

}

void main(){

int a[10000], num, pos, i, ch, ele;

clock\_t start,end;

double time;

printf("Enter number of elements: ");

scanf("%d", &num);

printf("Elements: ");

for(i=0;i<num;i++){

a[i]=(int)rand()%100;

printf("%d ",a[i]);

}

for(;;){

printf("\nEnter choice\n1.Linear search\n2.Binary search\n3.Exit\n");

scanf("%d", &ch);

switch(ch){

case 1: printf("Enter element to be searched: ");

scanf("%d",&ele);

start=clock();

pos=linear(ele,num,a,0);

end=clock();

time=((double)(end-start))/CLOCKS\_PER\_SEC;

printf("Time taken=%1f\n",time);

break;

case 2: sort(num,a);

printf("sorted list:\n");

for(i=0;i<num;i++)

printf("%d ",a[i]);

printf("\nEnter element to be searched: ");

scanf("%d",&ele);

start=clock();

pos=binary(ele,a,0,num-1);

end=clock();

time=((double)(end-start))/CLOCKS\_PER\_SEC;

printf("Time taken=%1f\n",time);

break;

case 3: exit(1);

default: printf("Wrong choice!\n");

}

if(pos==0)

printf("Element not found\n");

else

printf("Element found at position %d\n", pos);

}

}