

// linear and binary search calculate time taken for different values of n

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
#include<stdlib.h>
#include<time.h>
```

```
void linear(int n,int arr[])
{
    int item=200000;
    int flag=-1;

    for(int i=0;i<n;i++)
    {
        if(arr[i]==item)
        {
            flag=i+1;
            break;
        }
    }
    /*if(flag>0)
        printf("value found at position %d \n",flag);
    else
        printf("value not found\n");*/
}
```

```
void binary(int n,int arr[])
{
    int item=500000,flag=-1;

    int l=0,u=n-1,mid=(l+u)/2;

    while((l<u))
    {
        if(arr[mid]==item)
        {
            flag=mid+1;
            break;
        }
    }
```

```

        else if(arr[mid]<item)
            l=mid+1;
        else
            u=mid-1;

        mid=(l+u)/2;
    }

    /*if(flag>0)
        printf("value found at position %d\n",flag);
    if(flag== -1)
        printf("value not found\n");*/
}

void main()
{
    int ch,n,a[150000],b[150000];
    clock_t start,end;
    for(;;)
    {
        printf("1.linear search \n 2.binary seach \n else exit \n");
        scanf("%d",&ch);
        switch(ch)
        {
            case 1:n=5000;
                printf("time taken by linear search for different values of n: \n");
                while(n<=145000)
                {
                    for(int i=0;i<n;i++)
                        a[i]=i;

                    start=clock();
                    linear(n,a);
                    //delay
                    for(int j=0;j<=100;j++);

                    end=clock();

                    printf("time taken by %d elements = %f secs\n",n,((double)(end-start))/CLOCKS_PER_SEC);

```

```

        n=n+10000;
        }
        break;

        case 2:n=5000;
        printf("time taken by binary search for different values of n: \n");
        while(n<=145000)
        {
                for(int i=0;i<n;i++)
                        b[i]=i;

                start=clock();
                binary(n,b);
//delay
                for(int j=0;j<=100;j++);

                end=clock();

                printf("time taken by %d elements = %f secs
\n",n,((double)(end-start))/CLOCKS_PER_SEC);
                n=n+10000;
                }
                break;

                default:exit(0);
        }
}
}

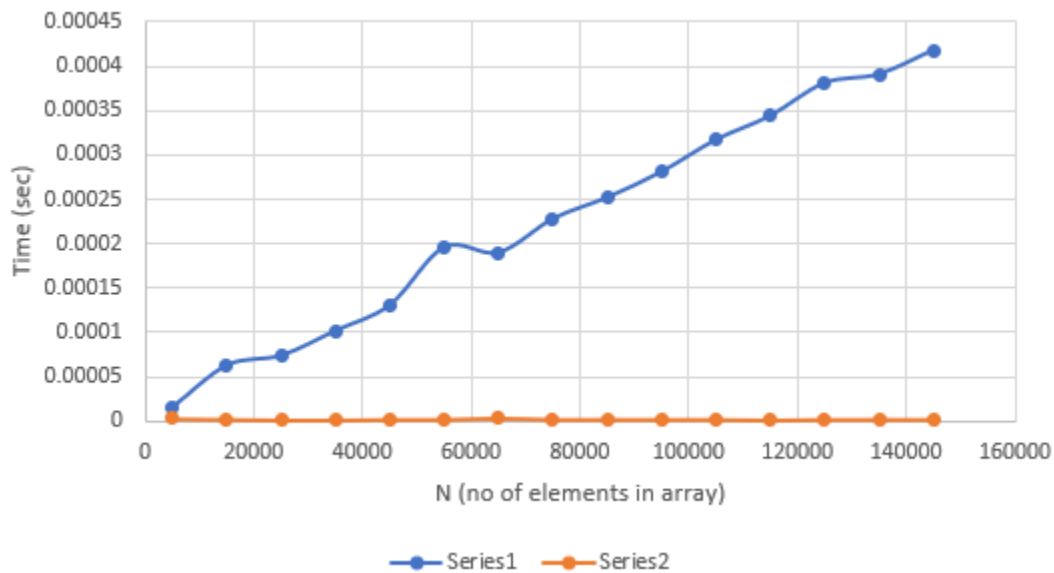
```

```
1.linear search
 2.binary seach
  else exit
1
time taken by linear search for different values of n:
time taken by 5000 elements = 0.000016 secs
time taken by 15000 elements = 0.000063 secs
time taken by 25000 elements = 0.000074 secs
time taken by 35000 elements = 0.000102 secs
time taken by 45000 elements = 0.000131 secs
time taken by 55000 elements = 0.000196 secs
time taken by 65000 elements = 0.000190 secs
time taken by 75000 elements = 0.000228 secs
time taken by 85000 elements = 0.000252 secs
time taken by 95000 elements = 0.000281 secs
time taken by 105000 elements = 0.000317 secs
time taken by 115000 elements = 0.000344 secs
time taken by 125000 elements = 0.000381 secs
time taken by 135000 elements = 0.000391 secs
time taken by 145000 elements = 0.000418 secs
```

```

1.linear search
2.binary seach
else exit
2
time taken by binary search for different values of n:
time taken by 5000 elements = 0.000003 secs
time taken by 15000 elements = 0.000002 secs
time taken by 25000 elements = 0.000001 secs
time taken by 35000 elements = 0.000001 secs
time taken by 45000 elements = 0.000002 secs
time taken by 55000 elements = 0.000002 secs
time taken by 65000 elements = 0.000004 secs
time taken by 75000 elements = 0.000002 secs
time taken by 85000 elements = 0.000002 secs
time taken by 95000 elements = 0.000002 secs
time taken by 105000 elements = 0.000002 secs
time taken by 115000 elements = 0.000001 secs
time taken by 125000 elements = 0.000002 secs
time taken by 135000 elements = 0.000002 secs
time taken by 145000 elements = 0.000002 secs

```



Here series 1 is linear search time curve
series 2 is binary search time curve