Name: - Atish Kumar

Roll No: - 120CS0173

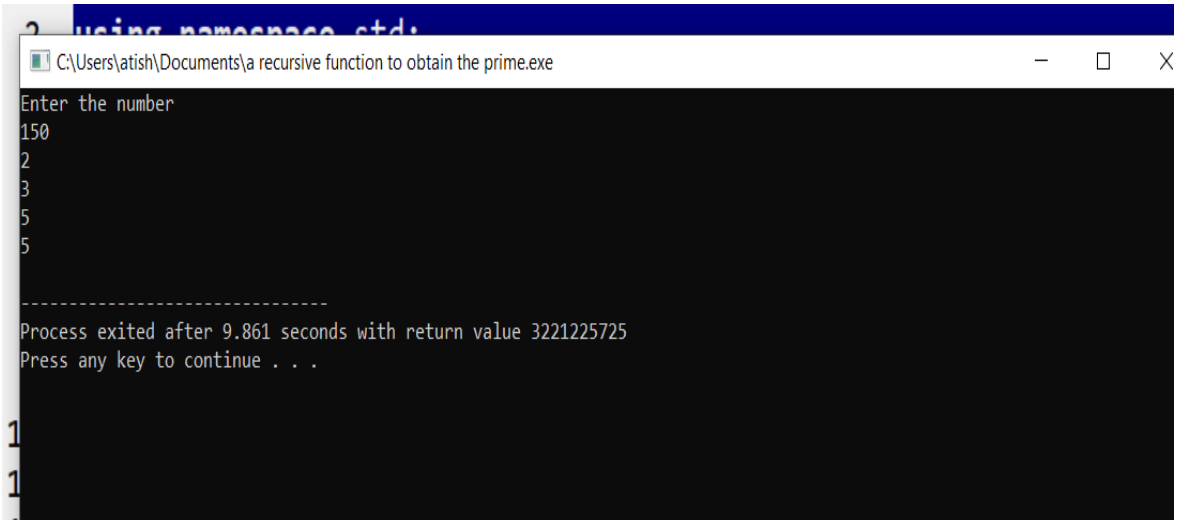
Lab Sheet: - 03

Q1. A positive integer is entered through the keyboard; write a recursive function to obtain the prime factors of that integer. Also, analyse the time and space complexity of your algorithm?

Program:-

```
#include<iostream>
using namespace std;
void prime(int n, int x){
if(n<1){
return;
}
else if(n%x==0){
cout<<x<<endl;
prime(n/x,x);
}
else{
prime(n, x+1);
}
}
int main(){
int n;
cout<<"Enter the number \n";
cin>>n;
prime(n, 2);
return 0;
}
```

Output:-



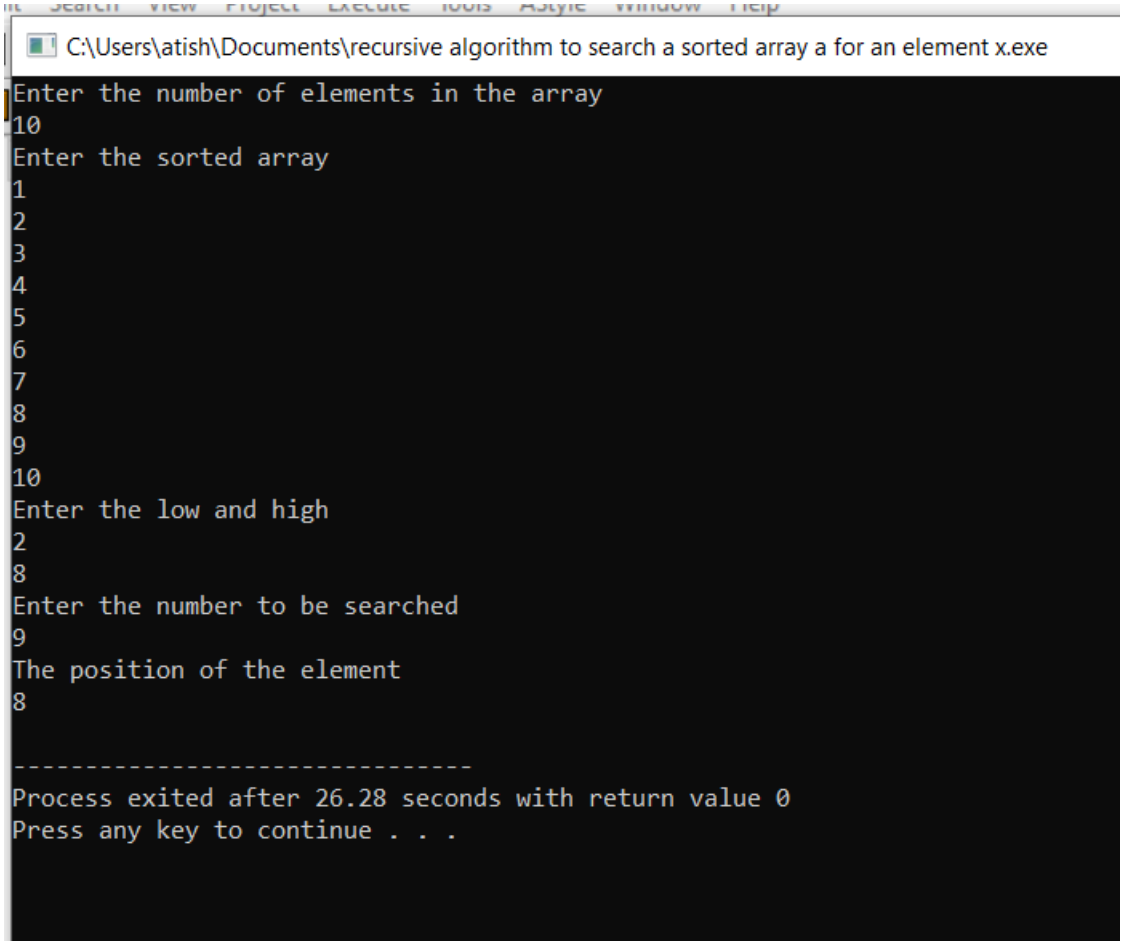
Q2. Design a recursive algorithm to search a sorted array a for an element x between a[low] and a[high]. If no element is found it returns -1. Also, analyse the time and space complexity of your algorithm.

Program:-

```
#include<iostream>
using namespace std;
void search(int a[], int x, int y, int sea){
int mid=(x+y)/2;
if(a[mid]==sea){
cout<<mid<<endl;
return;
}
else if(x==y){
cout<<-1<<endl;
return;
}
if(a[mid]>sea){
```

```
search(a, x, mid, sea);
}
else if(a[mid]<sea){
search(a, (mid+1), y, sea);
}
}
int main(){
int n;
cout<<"Enter the number of elements in the array \n";
cin>>n;
int a[n];
cout<<"Enter the sorted array \n";
for(int i=0;i<n;i++){
cin>>a[i];
}
int x,y;
cout<<"Enter the low and high \n";
cin>>x>>y;
int sea;
cout<<"Enter the number to be searched \n";
cin>>sea;
cout<<"The position of the element \n";
search(a, x, y, sea);
return 0;
}
```

Output:-


C:\Users\atish\Documents\recursive algorithm to search a sorted array a for an element x.exe
Enter the number of elements in the array
10
Enter the sorted array
1
2
3
4
5
6
7
8
9
10
Enter the low and high
2
8
Enter the number to be searched
9
The position of the element
8

Process exited after 26.28 seconds with return value 0
Press any key to continue . . .

Q3. WAP to implement the Linear linked list. Perform the following operations on the linked list:

- **Creating an empty linked list**
- **Adding the numbers at the beginning of the linked list**
- **Addition of numbers after a particular location.**
- **Counting the no of nodes.**
- **Displaying the linked list**.

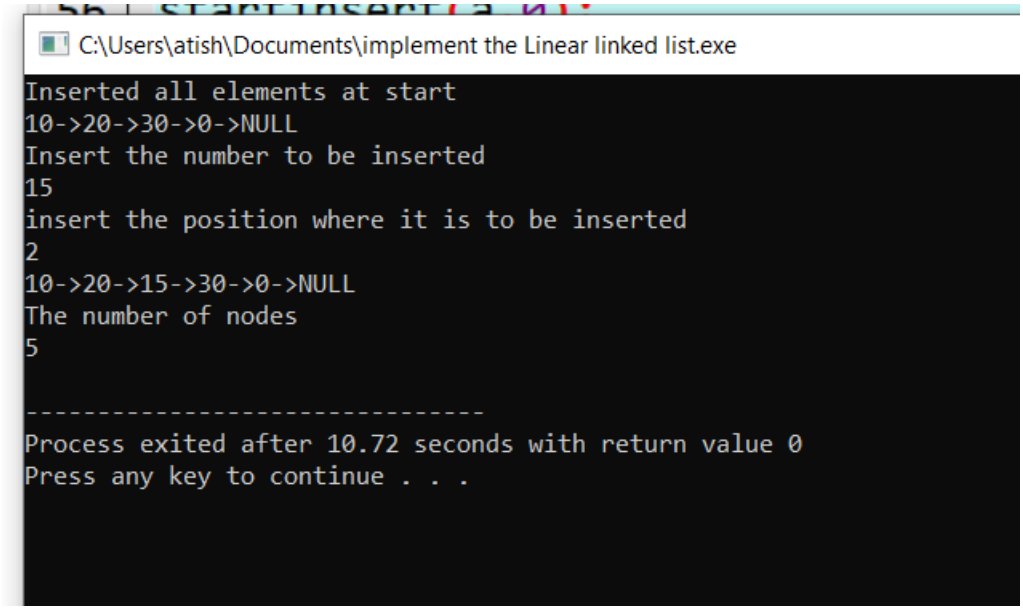
Program:-

```
#include<iostream>
using namespace std;
class node{
public:
int data;
node* next;
node(int d){
data=d;
next = NULL;
}
};
void startinsert(node* &head, int d){
if(head==NULL){
head = new node(d);
return;
}
node* n = new node(d);
n->next = head;
head=n;
}
```

```
void midinsert(node* &head, int d, int p){
if(p==0){
startinsert(head, d);
return;
}
node* n = new node(d);
node* temp = head;
for(int i=0;i<p-1;i++){
temp = temp->next;
}
n->next = temp->next;
temp->next = n;
return;
}
int nodecount(node* head){
int count = 0;
while(head!=NULL){
head=head->next;
count++;
}
return count;
}
void printnode(node* head){
if(head == NULL){
return;
}
```

```
while(head!=NULL){
cout<<(head->data)<<"->";
head=head->next;
}
cout<<"NULL\n";
return;
}
int main(){
node* a=NULL;
startinsert(a,0);
startinsert(a,30);
startinsert(a,20);
startinsert(a,10);
cout<<"Inserted all elements at start\n";
printnode(a);
int insertnum, insertpos;
cout<<"Insert the number to be inserted\n";
cin>>insertnum;
cout<<"insert the position where it is to be inserted\n";
cin>>insertpos;
midinsert(a, insertnum, insertpos);
printnode(a);
cout<<"The number of nodes\n";
cout<<nodecount(a)<<endl;
}
```

Output:-



Q4. Write a program that inserts 25 random integers from 0 to 100 in sorted order in a linked list. The program should calculate the sum of the elements and the floating-point average of the elements.

Example:

The list is: 6 12 14 20 27 31 31 34 37 38 56 59 63 66 72 73 73 76 77 79 88 94 95 96 97

*** The sum is: 1414 The average is: 56.560000**

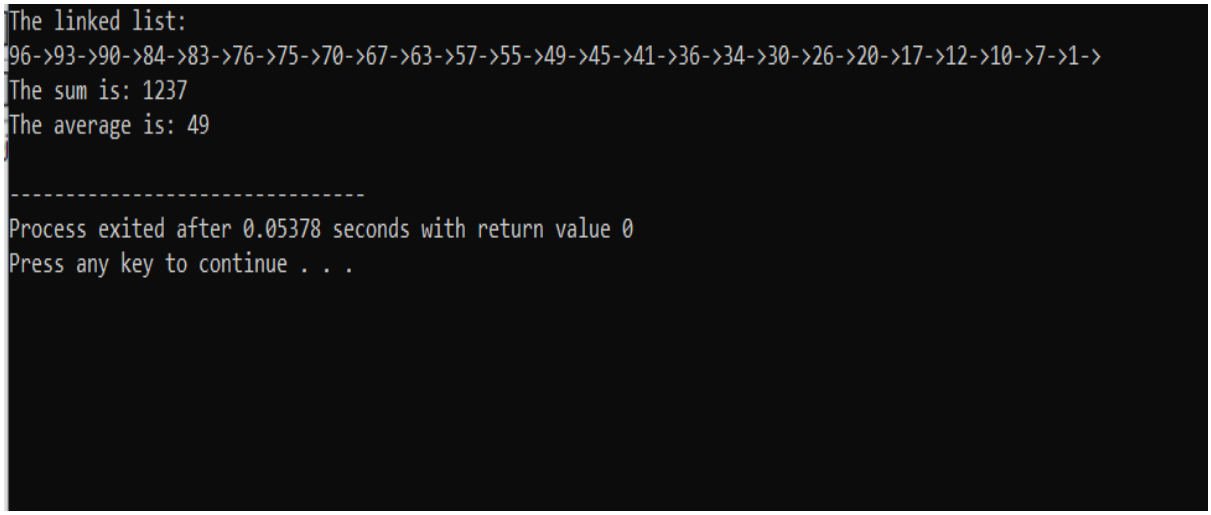
Program:-

```
#include<bits/stdc++.h>

using namespace std;

class node{

public:

int data;

node* next;

node(int d){
```

```
data = d;
next=NULL;
}
};
void insert(node* &head, int d){
if(head==NULL){
head=new node(d);
return;
}
node* n = new node(d);
n->next=head;
head = n;
}
void print(node* head){
int count = 0;
if(head == NULL){
return;
}
while(head!=NULL){
cout<<(head->data)<<"->";
head=head->next;
}
cout<<"\n";
return;
}
int sum(node* head){
```

```
int count = 0;
while(head!=NULL){
count+=head->data;
head=head->next;
}
return count;
}
int main(){
node* a = NULL;
int low=0;
int high=3;
for(int i=0;i<25;i++){
int num;
num = (rand()%(high-low+1))+low;
insert(a,num);
low+=4;
high+=4;
}
cout<<"The linked list: "<<endl;
print(a);
int nodesum = sum(a);
cout<<"The sum is: "<<nodesum<<endl;
cout<<"The average is: "<<(double)(nodesum/25)<<endl;
return 0;
}
```

Output:-


The linked list:
96->93->90->84->83->76->75->70->67->63->57->55->49->45->41->36->34->30->26->20->17->12->10->7->1->
The sum is: 1237
The average is: 49

Process exited after 0.05378 seconds with return value 0
Press any key to continue . . .