UNIVERSITI PENDIDIKAN SULTAN IDRIS

اونيۏرسيتي ڤندидikan سلطان ادريس

SULTAN IDRIS EDUCATION UNIVERSITY

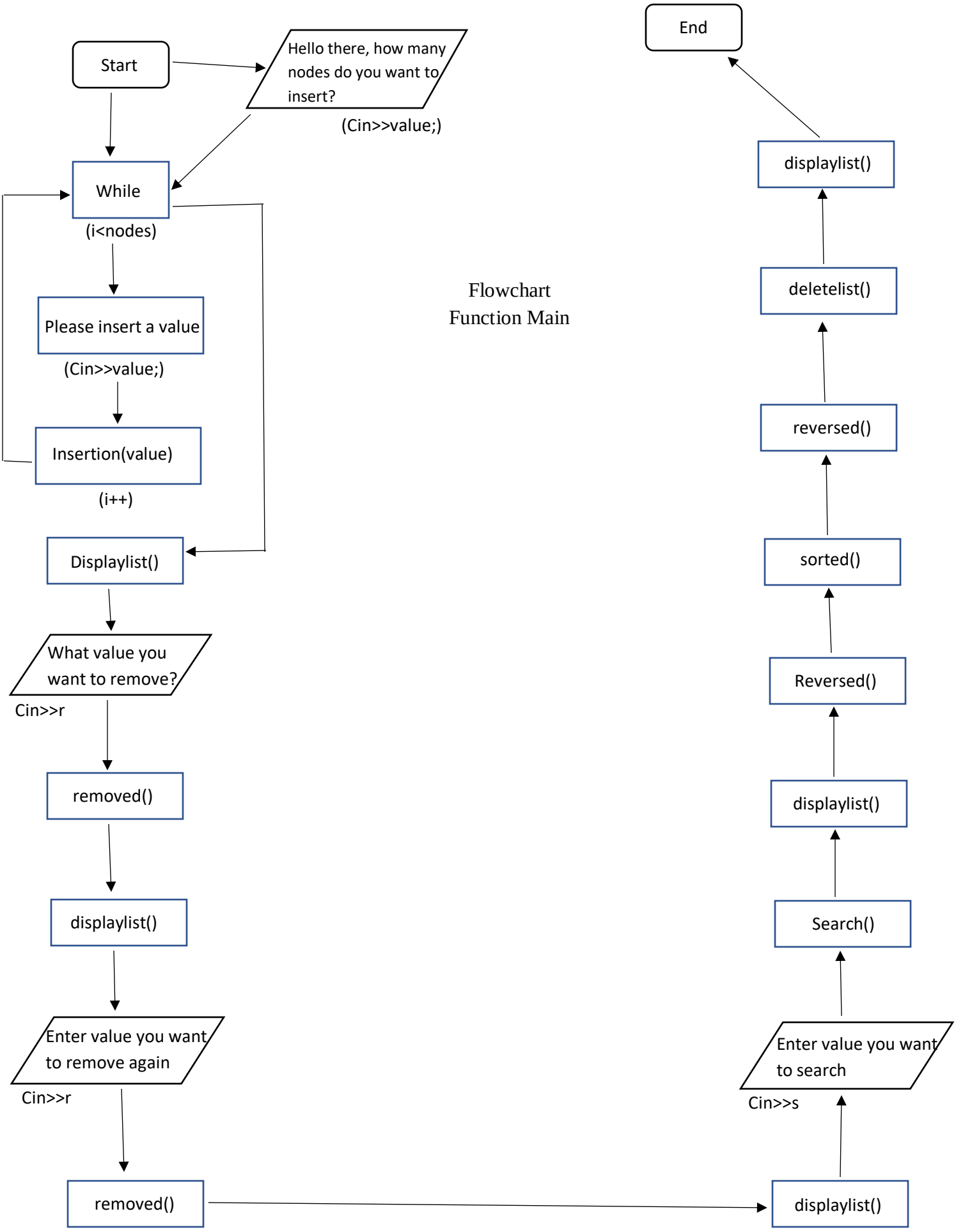
UNIVERSITI PENDIDIKAN SULTAN IDRIS
35900 TANJONG MALIM, PERAK.

MTS3023

DATA STRUCTURE

INDIVIDUAL ASSIGNMENT

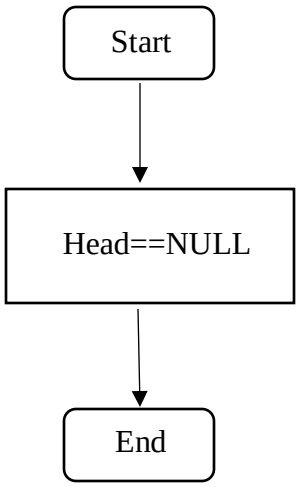
BIL	NAMA PELAJAR	NO. MATRIK
1	MOHD IZZUL IKHWAN BIN MOHD YUSOF	D20201095609
KUMPULAN KULIAH	E	
NAMA PENGAJAR	DR. Ramlah Binti Mailok	



Flowchart
Function Main

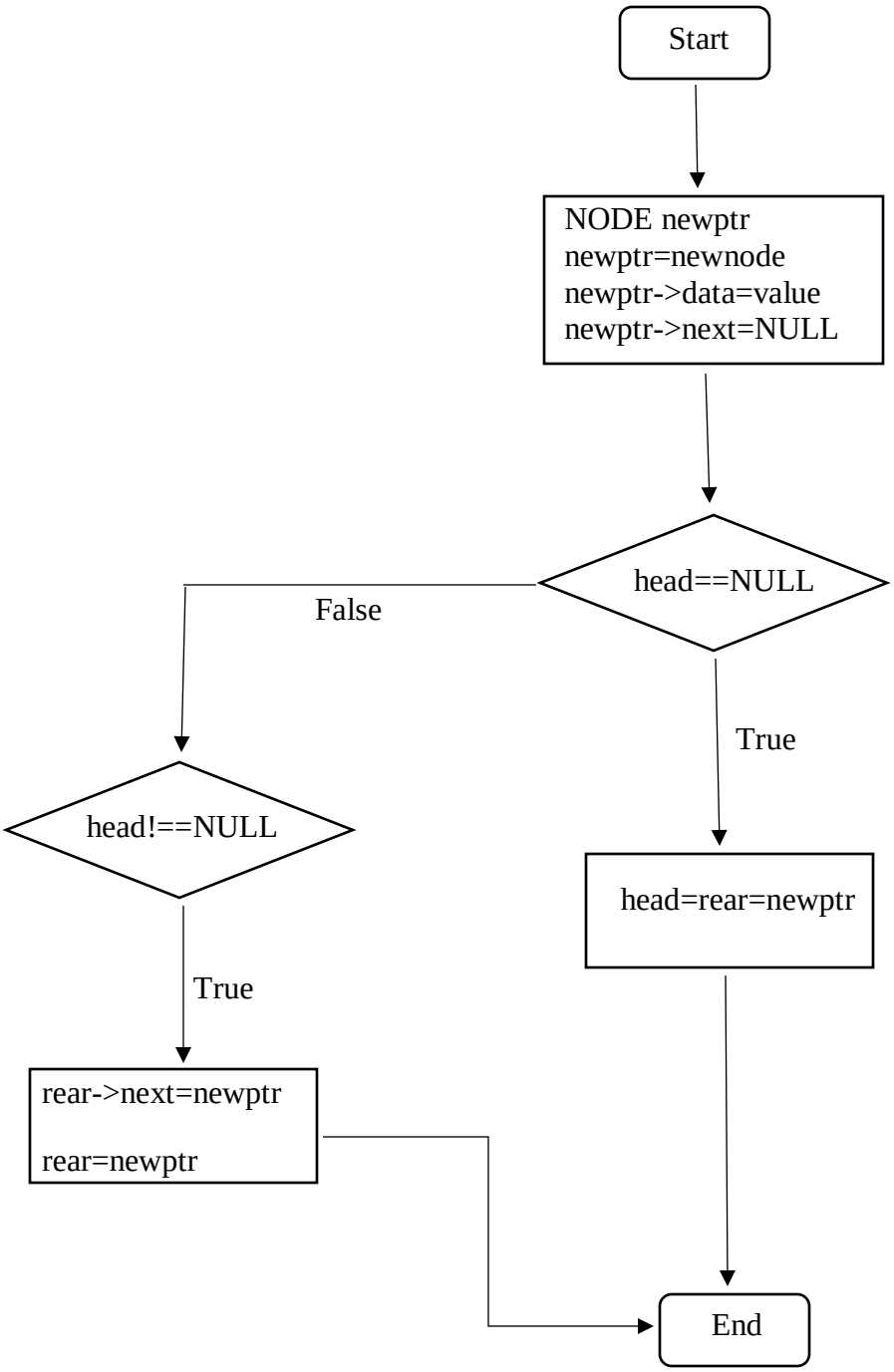
Flowchart

Function createnode()

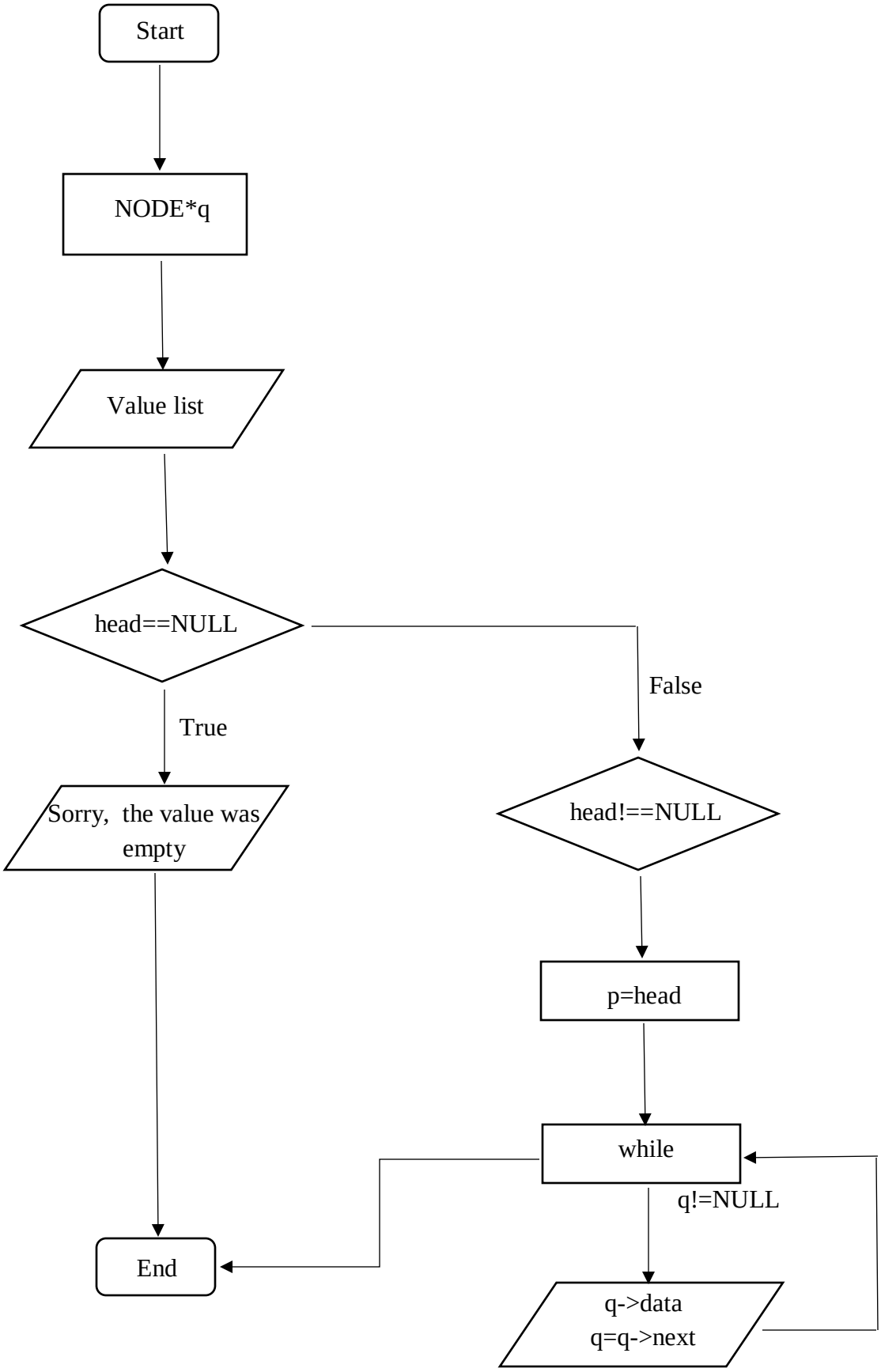


Flowchart

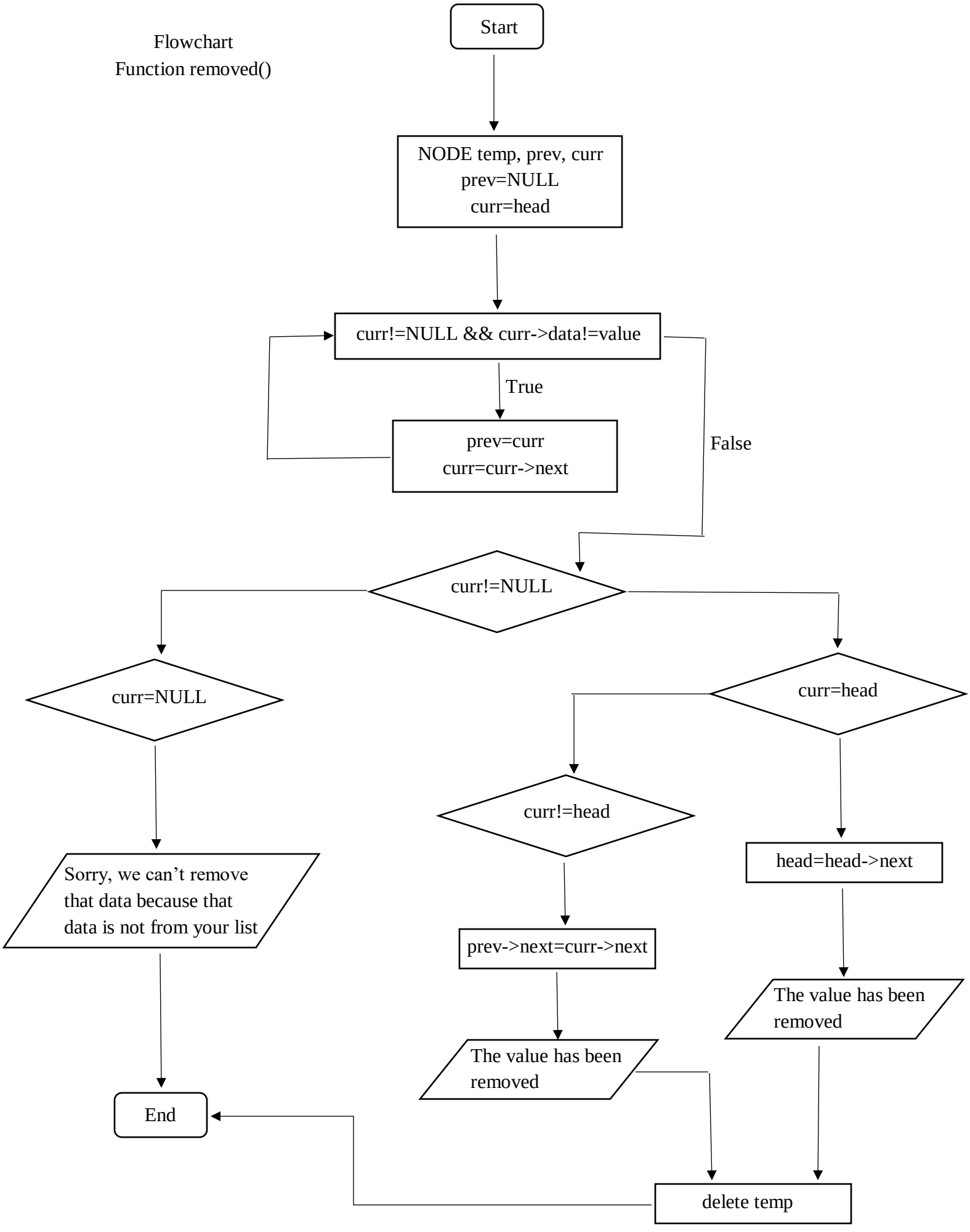
Function insertion(int value)



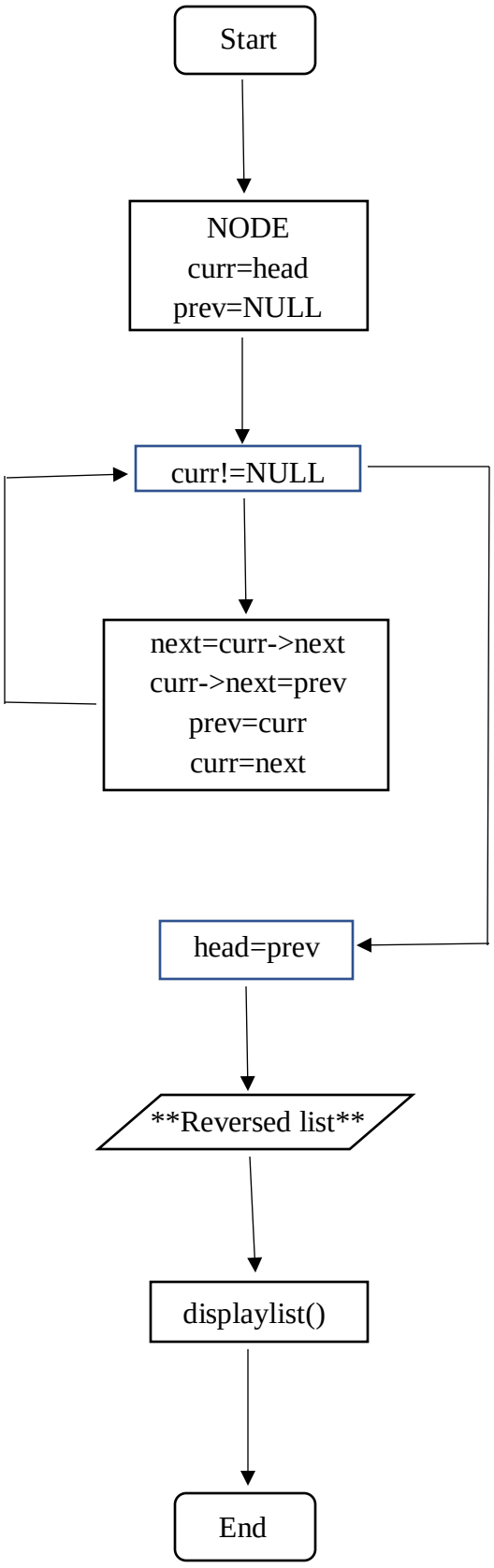
Flowchart
Function displaylist()



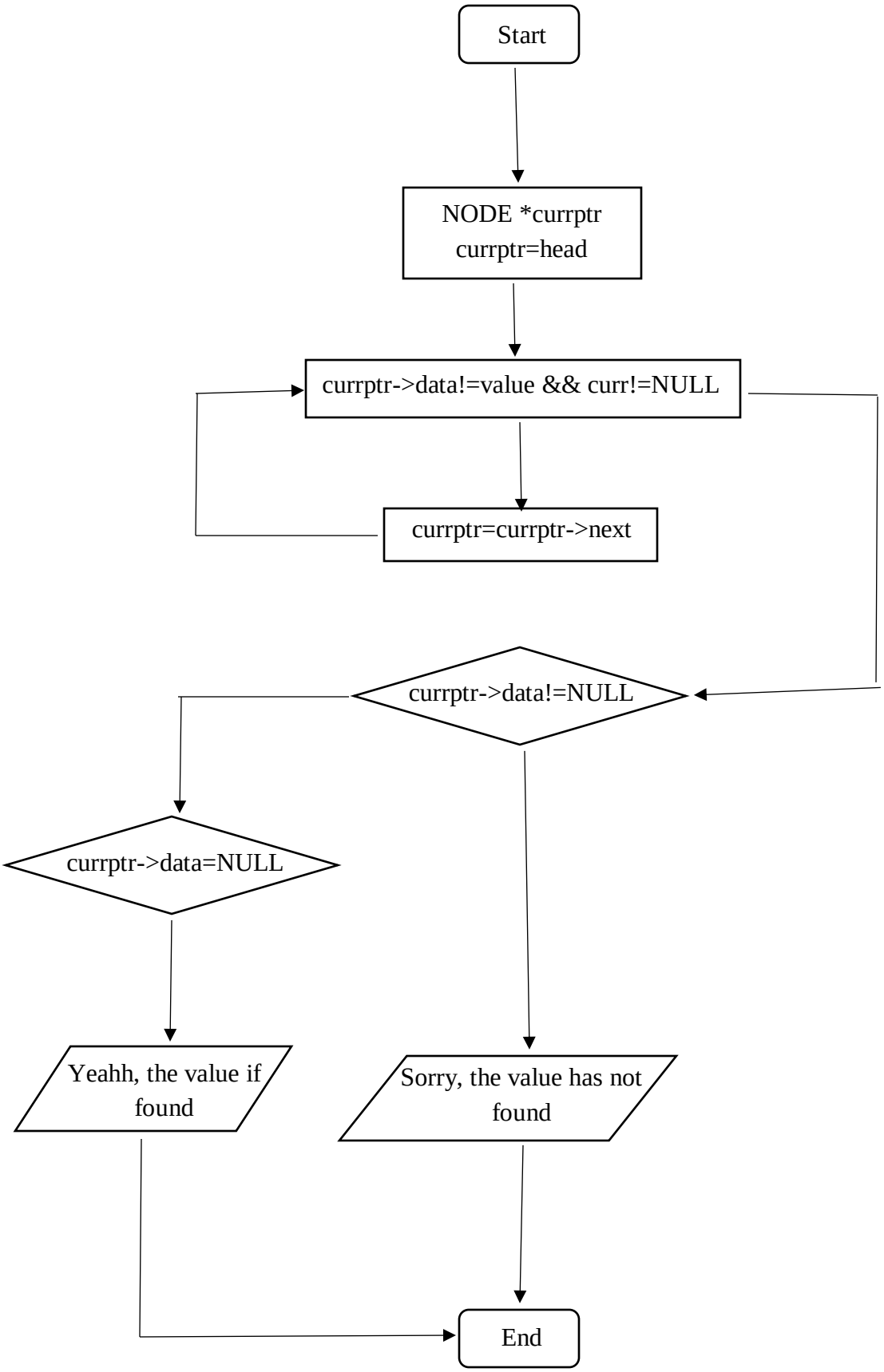
Flowchart
Function removed()



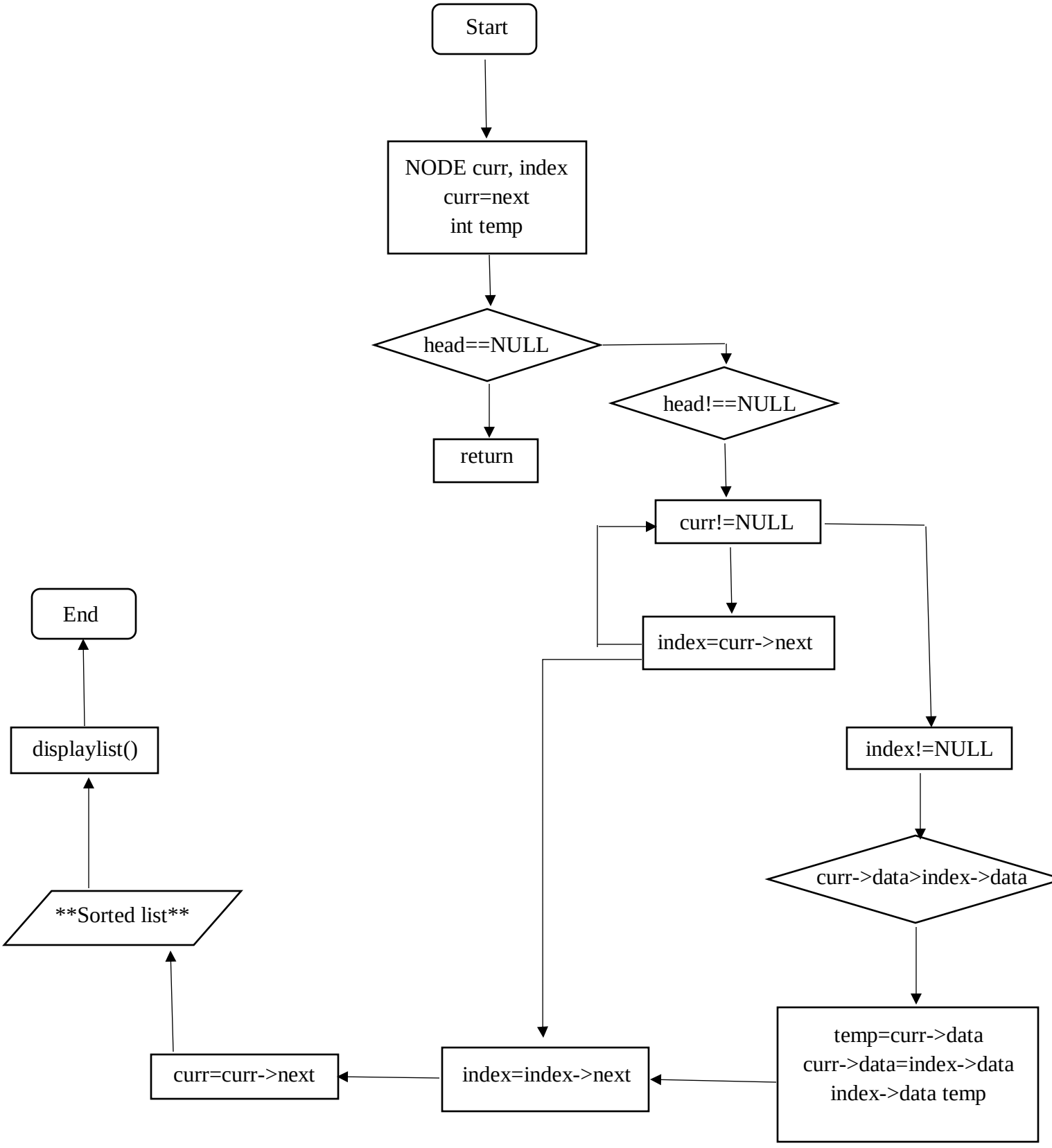
Flowchart
Function reversed()



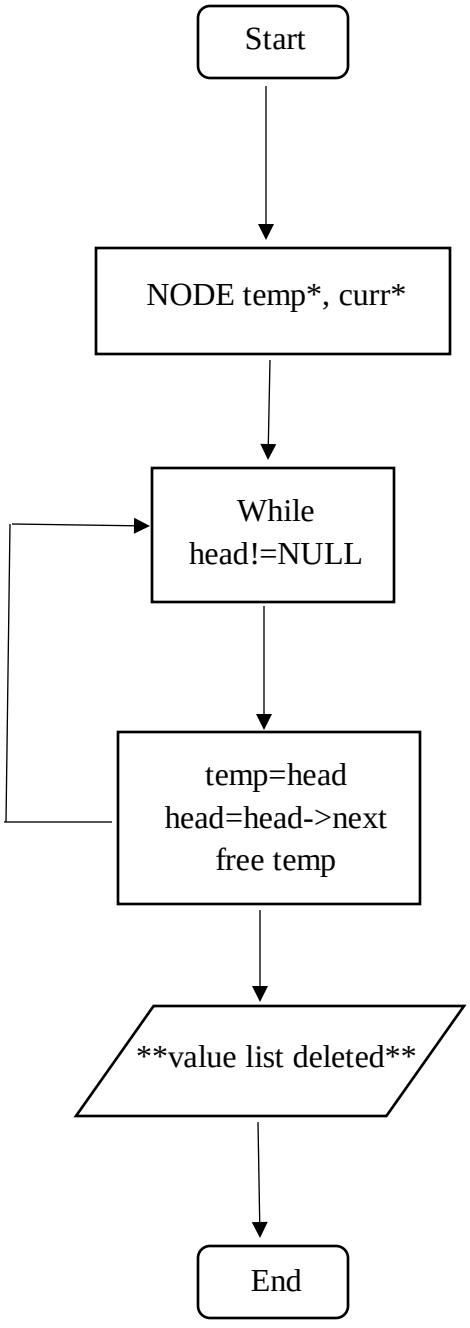
Flowchart
Function searching()



Flowchart
Function sorted()



Flowchart
Function deletelist()



Linked list programming (coding):

```
#include <iostream.h>
using namespace std;

struct NODE{
  int data;
  NODE *next;
};

  NODE *head;
  NODE *rear;

void cratenode(){
  head==NULL;
}

void insertion(int value){
  NODE *newptr;
  newptr=new NODE;
  newptr->data=value;
  newptr->next=NULL;

if (head==NULL)
  head=rear=newptr;
else{
  rear->next=newptr;
  rear=newptr;
 }
}

void displaylist(){
  NODE *q;

if(head==NULL)
cout<<"Sorry, the value was empty\n\n\n";
else{
  q=head;
while(q!=NULL){
cout<<q->data<<"  ";
```

```
  q=q->next;
 }
}
}


void removed(int value){
  NODE *temp, *prev, *curr;
  prev=NULL;
  curr=head;

while((curr!=NULL) && (curr->data!=value)){
  prev=curr;
  curr=curr->next;
}

  temp=curr;

if(curr!=NULL){
if(curr==head){
  head=head->next;
cout<<"\nValue has been removed!\n";
}
else{
  prev->next=curr->next;
cout<<"\nValue has been removed!\n";
}
}
else{
cout<<"\nSorry we can't remove that value because that value is
not from your value list\n";
}
  delete temp;
}

bool search(int value){
  NODE *currptr;
  currptr=head;
while((currptr->data!=value) && (currptr!=NULL)){
   currptr=currptr->next;
}
```

```
if(currptr->data!=value){
  cout<<"\nSorry, the value has not found\n";
return false;
}
else{
  cout<<"\nYeah, the value is here\n";
return true;
}
}

void reverse(){
  NODE *prev, *curr, *next;
  curr=head;
  prev=NULL;

while(curr!=NULL){
  next=curr->next;
  curr->next=prev;
  prev=curr;
  curr=next;
}
  head=prev;
cout<<"\n\n**Reverse list**\n";
displaylist();
}

void sorted(){
  NODE *curr, *index;
  curr=head;
int temp;
if(head==NULL){
  return;
}
else{
while(curr!=NULL){
  index=curr->next;

while(index!=NULL){
if(curr->data>index->data){
  temp=curr->data;
  curr->data=index->data;
```

```
  index->data=temp;
}
  index=index->next;
}
  curr=curr->next;
 }
}
cout<<"\n\n**Sorted list**\n";
displaylist();
}


void deletelist(){
  NODE *temp, *curr;

while (head!=NULL){
  temp=head;
  head=head->next;
  free(temp);
}
cout<<"\n\n**Value List Deleted**\n";
}

int main(){

int i = 0, nodes, value;

cout<<"\n----------------------------------------\n";
cout<<"\n           HELLO THERE :)\n";
cout<<"\n----------------------------------------\n";
cout<<"\nHow many nodes do you want to insert? : ";
cin>>nodes;

while(i < nodes){

cout<<"\nPlease enter a value: ";
cin>>value;
insertion(value);
i++;
}
```

```
cout<<"\n  **Value List**\n";
displaylist();

int r;
cout<<"\n\nEnter value you want to removed?: ";
cin>>r;
removed(r);

cout<<"\n **Value List**\n";
displaylist();

cout<<"\n\nEnter the value you want to remove again: ";
cin>>r;
removed(r);

cout<<"\n **Value List**\n";
displaylist();

int s;
cout<<"\n\nPlease enter value that you want to search: ";
cin>>s;
search(s);

cout<<"\n **Value List**\n";
displaylist();

reverse();
sorted();
reverse();
deletelist();

displaylist();
}
```

Linked list programming (output):

```
---------------------------------------

          HELLO THERE :)

---------------------------------------
How many nodes do you want to insert? : 5
Please enter a value: 11
Please enter a value: 22
Please enter a value: 33
Please enter a value: 44
Please enter a value: 55
  **Value List**
11  22  33  44  55
Enter value you want to removed?: 11

Value has been removed!

 **Value List**
22  33  44  55
Enter the value you want to remove again: 66

Sorry we can't remove that value because that value is not from your value list

 **Value List**
22  33  44  55
Please enter value that you want to search: 33

Yeah, the value is here

 **Value List**
22  33  44  55

**Reverse list**
55  44  33  22

**Sorted list**
22  33  44  55

**Reverse list**
55  44  33  22

**Value List Deleted**
Sorry, the value was empty



 ----jGRASP: operation complete.
```