

ASSIGNMENT-2

NAME – ANSHIKA

ROLL NO. – 102003183

SUBGROUP – 2COE8

Question 1-

```
#include<iostream>
```

```
#include<cstdio>
```

```
#include<cstdlib>
```

```
using namespace std;
```

```
struct node{
```

```
    int info;
```

```
    struct node *next;
```

```
}*start;
```

```
class list{
```

```
    public:
```

```
        node* createNode(int);
```

```
        void insertBegin();
```

```
        void insertEnd();
```

```
        void insertBetween();
```

```
        void delBegin();
```

```
        void delEnd();
```

```
        void delBetween();
```

```
        void search();
```

```
        void display();
```

```
        list(){
```

```
            start=NULL;
```

```
        }
```

```
};
```

```

node *list::createNode(int val){
    struct node *temp, *s;
    temp=new(struct node);
    if(temp==NULL){
        cout<<"Memory not allocated"<<endl;
        return 0;
    }
    else{
        temp->info=val;
        temp->next=NULL;
        return temp;
    }
}

```

```

void list::insertBegin(){
    int val;
    cout<<"Enter the value to be inserted: ";
    cin>>val;
    struct node *temp, *ptr;
    temp=createNode(val);
    if(start==NULL){
        start=temp;
        start->next=NULL;
    }
    else{

```

```

        ptr=start;
        start=temp;
        start->next=ptr;
    }
    cout<<"Element inserted"<<endl;
}

```

```

void list::insertEnd(){
    int val;
    cout<<"Enter the value to be inserted: ";
    cin>>val;
    struct node *temp, *ptr, *s;
    temp=createNode(val);
    s=start;
    while(s->next!=NULL){
        s=s->next;
    }
    temp->next=NULL;
    s->next=temp;
    cout<<"Element inserted"<<endl;
}

```

```

void list::insertBetween(){
    int val, pos, counter=0;
    cout<<"Enter the value to be inserted: ";
    cin>>val;

```

```
struct node *temp, *ptr, *s;

temp=createNode(val);

cout<<"Enter the position at which node is to be inserted: ";

cin>>pos;

s=start;

while(s!=NULL){

    s=s->next;

    counter++;

}

if(pos==1){

    if(start==NULL){

        start=temp;

        start->next=NULL;

    }

    else{

        ptr=start;

        start=temp;

        start->next=ptr;

    }

}

else if(pos>1 && pos<=counter){

    s=start;

    for(int i=1; i<pos; i++){

        ptr=s;

        s=s->next;

    }
```

```
        ptr->next=temp;
        temp->next=s;
    }
    else{
        cout<<"Position out of range"<<endl;
    }
    cout<<"Element inserted"<<endl;
}
```

```
void list::delBegin(){
    struct node *temp;
    if(start==NULL){
        cout<<"List is empty"<<endl;
    }
    else{
        temp=start;
        start=start->next;
        free(temp);
    }
    cout<<"Element deleted"<<endl;
}
```

```
void list::delEnd(){
    struct node *temp, *prev;
    if(start==NULL){
        cout<<"List is empty"<<endl;
```

```

    }
    else{
        temp=start;
        while(temp->next!=0){
            prev=temp;
            temp=temp->next;
        }
        free(temp);
        prev->next=0;
    }
    cout<<"Element deleted"<<endl;
}

```

```

void list::delBetween(){
    int pos, counter=0;
    if(start==NULL){
        cout<<"List is empty"<<endl;
    }
    cout<<"Enter the position of the element to be deleted: ";
    cin>>pos;
    struct node *s, *ptr;
    s=start;
    if(pos==1){
        start=s->next;
    }
    else{

```

```

        while(s!=NULL){
            s=s->next;
            counter++;
        }
        if(pos>0 && pos<=counter){
            s=start;
            for(int i=1; i<pos; i++){
                ptr=s;
                s=s->next;
            }
            ptr->next=s->next;
        }
        else{
            cout<<"Position out of range"<<endl;
        }
        free(s);
        cout<<"Element deleted"<<endl;
    }
}

```

```

void list::search(){
    struct node *ptr;
    int val;
    cout<<"Enter the value whose position is to be displayed: ";
    cin>>val;
    ptr=start;

```



```

int index=0;
while(ptr!=NULL){
    if(ptr->info==val){
        cout<<"Element is at index "<<index<<endl;
        return;
    }
    ptr=ptr->next;
    index++;
}
cout<<"Element not found"<<endl;;
}

```

```

void list::display(){
    struct node *ptr;
    ptr=start;
    while(ptr!=NULL){
        cout<<ptr->info<<" ";
        ptr=ptr->next;
    }
    cout<<endl;
}

```

```

int main(){
    int choice, nodes, element, pos;
    list l;
    start=NULL;

```

cout<<"Enter\n1 to Insert at beginning,\n2 to Insert at end,\n3 to Insert in between,\n4 to Delete from beginning,\n5 to Delete from end,\n6 to Delete a specific node,\n7 to Search for a node and display its position from head,\n8 to Display all the node values,\n9 to Exit\n";

```
cin>>choice;
```

```
while(choice!=9){
```

```
switch(choice)
```

```
{
```

```
    case 1 : cout<<"Inserting node at the beginning"<<endl;
```

```
              l.insertBegin();
```

```
              cout<<endl;
```

```
              break;
```

```
    case 2 : cout<<"Inserting node at the end"<<endl;
```

```
              l.insertEnd();
```

```
              cout<<endl;
```

```
              break;
```

```
    case 3 : cout<<"Inserting node in between"<<endl;
```

```
              l.insertBetween();
```

```
              cout<<endl;
```

```
              break;
```

```
    case 4 : cout<<"Deleting node at the beginning"<<endl;
```

```
              l.delBegin();
```

```
              cout<<endl;
```

```
              break;
```

```
    case 5 : cout<<"Deleting node at the end"<<endl;
```

```
              l.delEnd();
```

```
              cout<<endl;
```

```
              break;
```

```

        case 6 : cout<<"Deleting node in between"<<endl;
                    l.delBetween();
                    cout<<endl;
                    break;

        case 7 : cout<<"Displaying position of searched element"<<endl;
                    l.search();
                    cout<<endl;
                    break;

        case 8 : cout<<"Displaying all the node values"<<endl;
                    l.display();
                    cout<<endl;
                    break;

        case 9 : return 0;
    }

    cout<<"Enter\n1 to Insert at beginning,\n2 to Insert at end,\n3 to Insert in between,\n4
to Delete from beginning,\n5 to Delete from end,\n6 to Delete a specific node,\n7 to Search for
a node and display its position from head,\n8 to Display all the node values,\n9 to Exit\n";

    cin>>choice;

}

}

```

C:\Users\hp\Documents\MenuDrivenSinglyLinkedList.exe

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
```

```
1
Inserting node at the beginning
Enter the value to be inserted: 2
Element inserted
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
```

```
2
Inserting node at the end
Enter the value to be inserted: 10
Element inserted
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
```

```
3
Inserting node in between
Enter the value to be inserted: 4
Enter the position at which node is to be inserted: 1
```

C:\Users\hp\Documents\MenuDrivenSinglyLinkedList.exe

```
Enter the value to be inserted: 4
Enter the position at which node is to be inserted: 1
Element inserted
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
```

```
3
Inserting node in between
Enter the value to be inserted: 6
Enter the position at which node is to be inserted: 2
Element inserted
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
```

```
3
Inserting node in between
Enter the value to be inserted: 8
Enter the position at which node is to be inserted: 3
Element inserted
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
```

Activate Windows
Go to Settings to activate Windows.

Activate Windows
Go to Settings to activate Windows.

C:\Users\hp\Documents\MenuDrivenSinglyLinkedList.exe

```
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
8
Displaying all the node values
4 6 8 2 10
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
4
Deleting node at the beginning
Element deleted
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
5
Deleting node at the end
Element deleted
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
```

Activate Windows
Go to Settings to activate Windows.

C:\Users\hp\Documents\MenuDrivenSinglyLinkedList.exe

```
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
6
Deleting node in between
Enter the position of the element to be deleted: 2
Element deleted
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
8
Displaying all the node values
6 2
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
6 to Delete a specific node,
7 to Search for a node and display its position from head,
8 to Display all the node values,
9 to Exit
7
Displaying position of searched element
Enter the value whose position is to be displayed: 2
Element is at index 1
```

```
Enter
1 to Insert at beginning,
2 to Insert at end,
3 to Insert in between,
4 to Delete from beginning,
5 to Delete from end,
```

Activate Windows
Go to Settings to activate Windows.

C:\Users\hp\Documents\MenuDrivenSinglyLinkedList.exe

```
3 to Insert in between,  
4 to Delete from beginning,  
5 to Delete from end,  
6 to Delete a specific node,  
7 to Search for a node and display its position from head,  
8 to Display all the node values,  
9 to Exit  
8  
Displaying all the node values  
6 2
```

Enter

```
1 to Insert at beginning,  
2 to Insert at end,  
3 to Insert in between,  
4 to Delete from beginning,  
5 to Delete from end,  
6 to Delete a specific node,  
7 to Search for a node and display its position from head,  
8 to Display all the node values,  
9 to Exit  
7
```

Displaying position of searched element

Enter the value whose position is to be displayed: 2

Element is at index 1

Enter

```
1 to Insert at beginning,  
2 to Insert at end,  
3 to Insert in between,  
4 to Delete from beginning,  
5 to Delete from end,  
6 to Delete a specific node,  
7 to Search for a node and display its position from head,  
8 to Display all the node values,  
9 to Exit  
9
```

Process exited after 167.2 seconds with return value 0

Press any key to continue . . . █

Activate Windows
Go to Settings to activate Windows.

Question 2-

```
#include<iostream>
```

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

```
using namespace std;
```

```
class Node{
```

```
    public:
```

```
        int data;
```

```
        Node* next;
```

```
};
```

```
Node* push (Node* head, int new_data){
```

```
    Node* new_node=new Node();
```

```
    new_node->data=new_data;
```

```
    new_node->next=head;
```

```
    head=new_node;
```

```
    return head;
```

```
}
```

```
int count(Node* head, int key){
```

```
    Node* current=head;
```

```
    int count=0;
```

```
    while(current!=NULL){
```

```
        if(current->data==key){
```

```
            count++;
```

```
        }
```

```
        current=current->next;
    }
    return count;
}
```

```
Node* deleteKey(Node *head, int key){
    Node *temp=head;
    while(head->data==key){
        head=head->next;
    }
    while(temp->next!=NULL){
        if(temp->next->data==key){
            temp->next=temp->next->next;
        }
        else{
            temp=temp->next;
        }
    }
    return head;
}
```

```
void printList(Node* node){
    while(node->next!=NULL){
        cout<<node->data<<" ";
        node=node->next;
    }
}
```



```

        cout<<node->data;
    }

int main(){
    Node* head=NULL;
    head=push(head, 1);
    head=push(head, 2);
    head=push(head, 1);
    head=push(head, 2);
    head=push(head, 1);
    head=push(head, 3);
    head=push(head, 1);
    cout<<"Created Linked List:\n";
    printList(head);
    int key;
    cout<<"\nEnter the key to be deleted: ";
    cin>>key;
    cout<<"Count of "<<key<<" is "<<count(head, 1);
    head=deleteKey(head, key);
    cout<<"\nLinked List after Deletion:\n";
    printList(head);
    return 0;
}

```

```
C:\Users\hp\Documents\CountandDeleteKey.exe
Created Linked List:
1 3 1 2 1 2 1
Enter the key to be deleted: 1
Count of 1 is 4
Linked List after Deletion:
3 2 2
-----
Process exited after 2.328 seconds with return value 0
Press any key to continue . . .
```

Question 3-

```
#include<iostream>
```

```
using namespace std;
```

```
class Node{
```

```
    public:
```

```
        int data;
```

```
        Node *next;
```

```
};
```

```
class NodeOperation{
```

```
    public:
```

```
        void pushNode(class Node** head_ref, int data){
```

```
            class Node *new_node=new Node();
```

```
            new_node->data=data;
```

```
            new_node->next=*head_ref;
```

```
            *head_ref=new_node;
```

```
        }
```

```
        void printNode(class Node *head){
```

```
            while(head!=NULL){
```

```
                cout<<head->data<<"->";
```

```
                head=head->next;
```

```
            }
```

```
            cout<<"NULL"<<endl;
```

```
        }
```

```
        void printMiddle(class Node *head){
```

```

        struct Node *slow_ptr=head;

        struct Node *fast_ptr=head;

        if(head!=NULL){

            while(fast_ptr!=NULL && fast_ptr->next!=NULL){

                fast_ptr=fast_ptr->next->next;

                slow_ptr=slow_ptr->next;

            }

            cout<<"The middle element is ["<<slow_ptr->data<<"]"<<endl;

        }

    }

};

int main(){

    class Node *head=NULL;

    class NodeOperation *temp=new NodeOperation();

    for(int i=5; i>0; i--){

        temp->pushNode(&head, i);

        temp->printNode(head);

        temp->printMiddle(head);

    }

    return 0;

}

```

```
C:\Users\hp\Documents\MiddleofLinkedList.exe
5->NULL
The middle element is [5]
4->5->NULL
The middle element is [5]
3->4->5->NULL
The middle element is [4]
2->3->4->5->NULL
The middle element is [4]
1->2->3->4->5->NULL
The middle element is [3]

-----
Process exited after 0.2725 seconds with return value 0
Press any key to continue . . .
```

Question 4-

```
#include<iostream>
```

```
using namespace std;
```

```
struct node{
```

```
    int data;
```

```
    struct node* next;
```

```
    node(int data){
```

```
        this->data=data;
```

```
        next=NULL;
```

```
    }
```

```
};
```

```
struct LinkedList{
```

```
    node* head;
```

```
    LinkedList(){
```

```
        head=NULL;
```

```
    }
```

```
    void reverse(){
```

```
        node* current=head;
```

```
        node *prev=NULL, *next=NULL;
```

```
        while(current!=NULL){
```

```
            next=current->next;
```

```
            current->next=prev;
```

```
            prev=current;
```

```
            current=next;
```

```

        }
        head=prev;
    }
    void print(){
        struct node* temp=head;
        while(temp!=NULL){
            cout<<temp->data<<" ";
            temp=temp->next;
        }
    }
    void push(int data){
        node* temp=new node(data);
        temp->next=head;
        head=temp;
    }
};

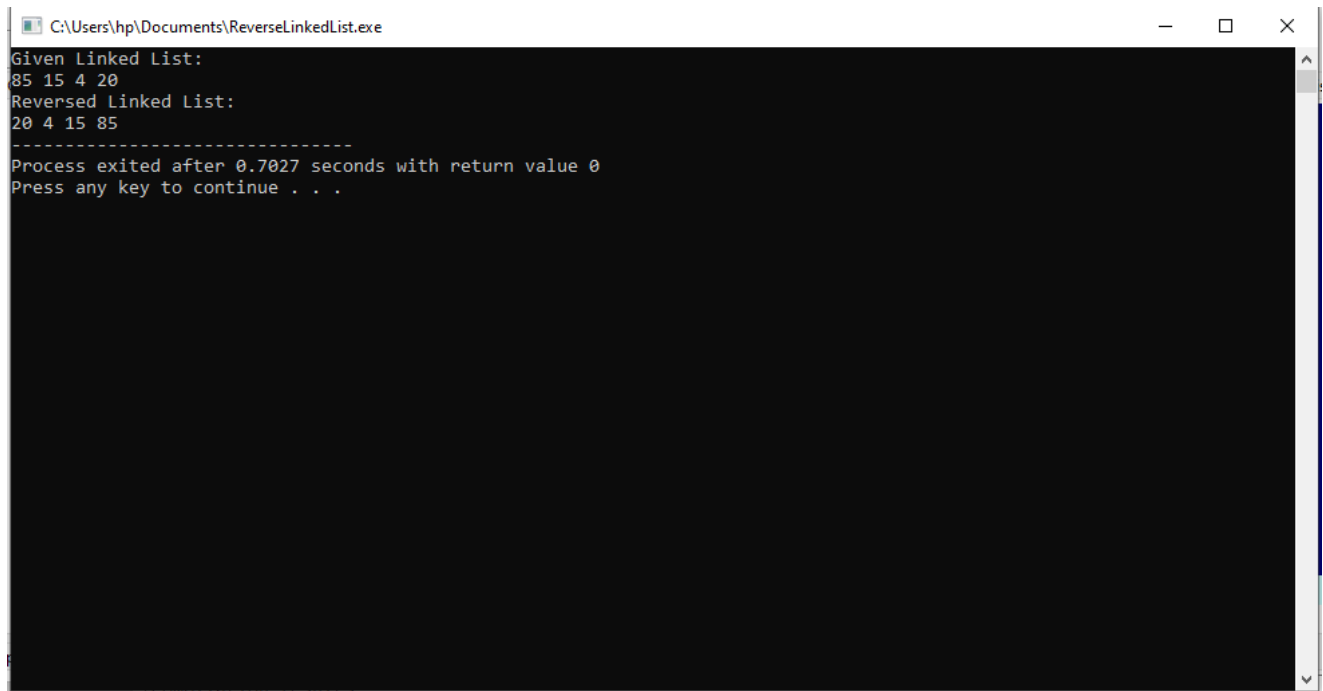
```

```

int main(){
    LinkedList ll;
    ll.push(20);
    ll.push(4);
    ll.push(15);
    ll.push(85);
    cout<<"Given Linked List:\n";
    ll.print();
    ll.reverse();
}

```

```
    cout<<"\nReversed Linked List:\n";  
  
    ll.print();  
  
    return 0;  
}
```



The screenshot shows a Windows command prompt window titled "C:\Users\hp\Documents\ReverseLinkedList.exe". The output of the program is as follows:

```
Given Linked List:  
85 15 4 20  
Reversed Linked List:  
20 4 15 85  
-----  
Process exited after 0.7027 seconds with return value 0  
Press any key to continue . . .
```

The window has a standard Windows title bar with minimize, maximize, and close buttons. A vertical scrollbar is visible on the right side of the command prompt area.