FAIZAN CHOUDHARY

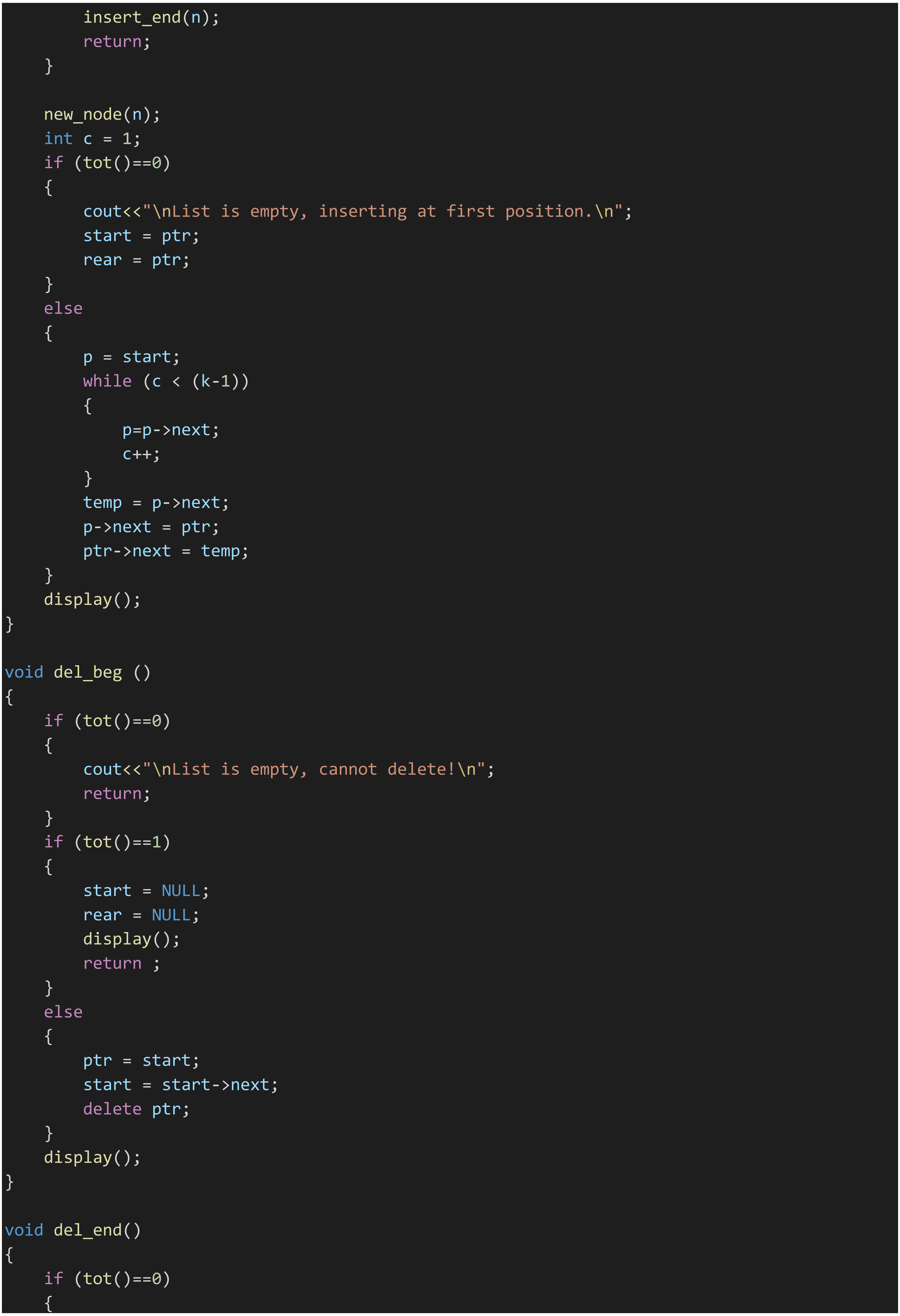
20BCS021

DSA LAB

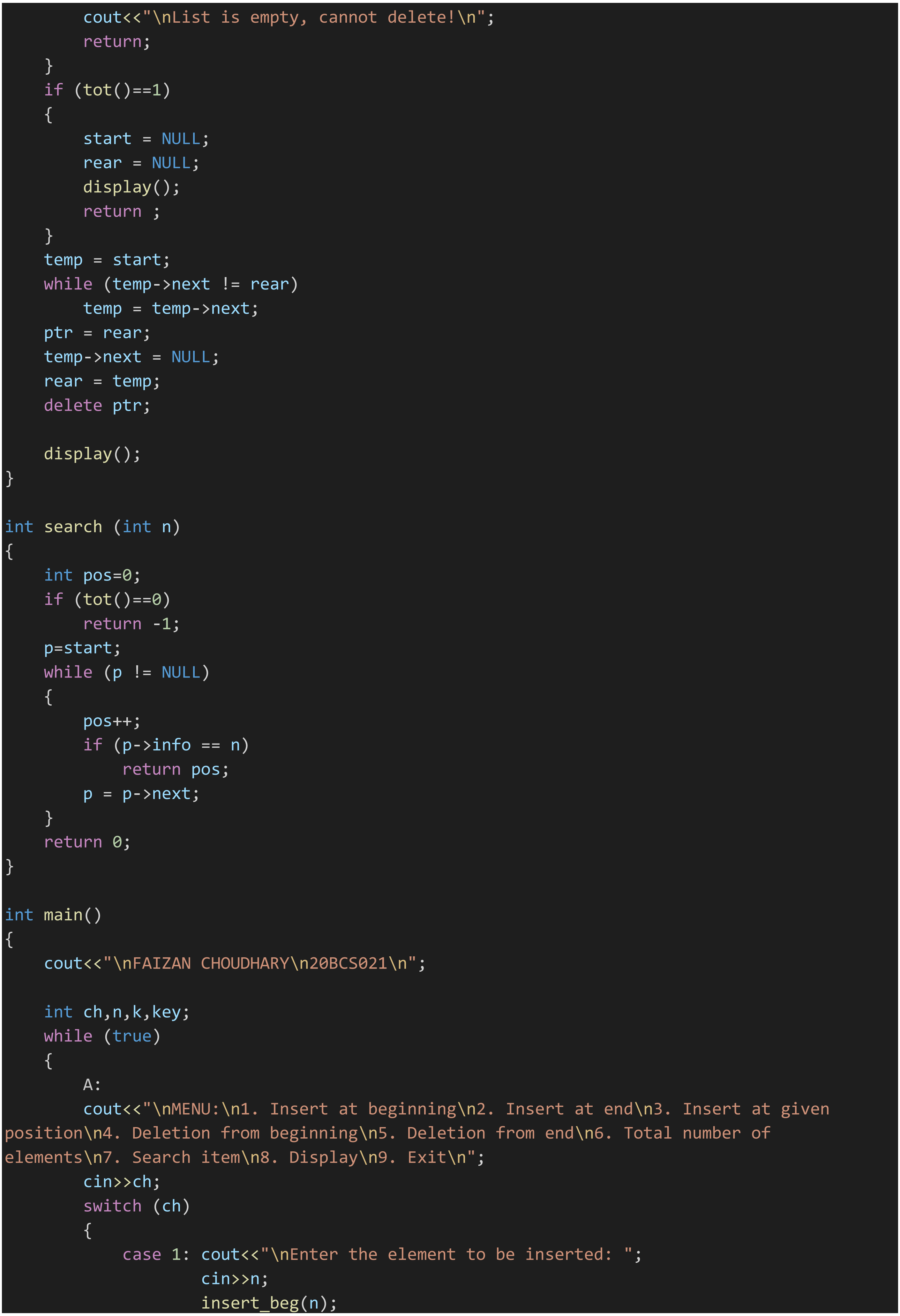
7th December 2021

CODE: (code pasted in this format for readability)

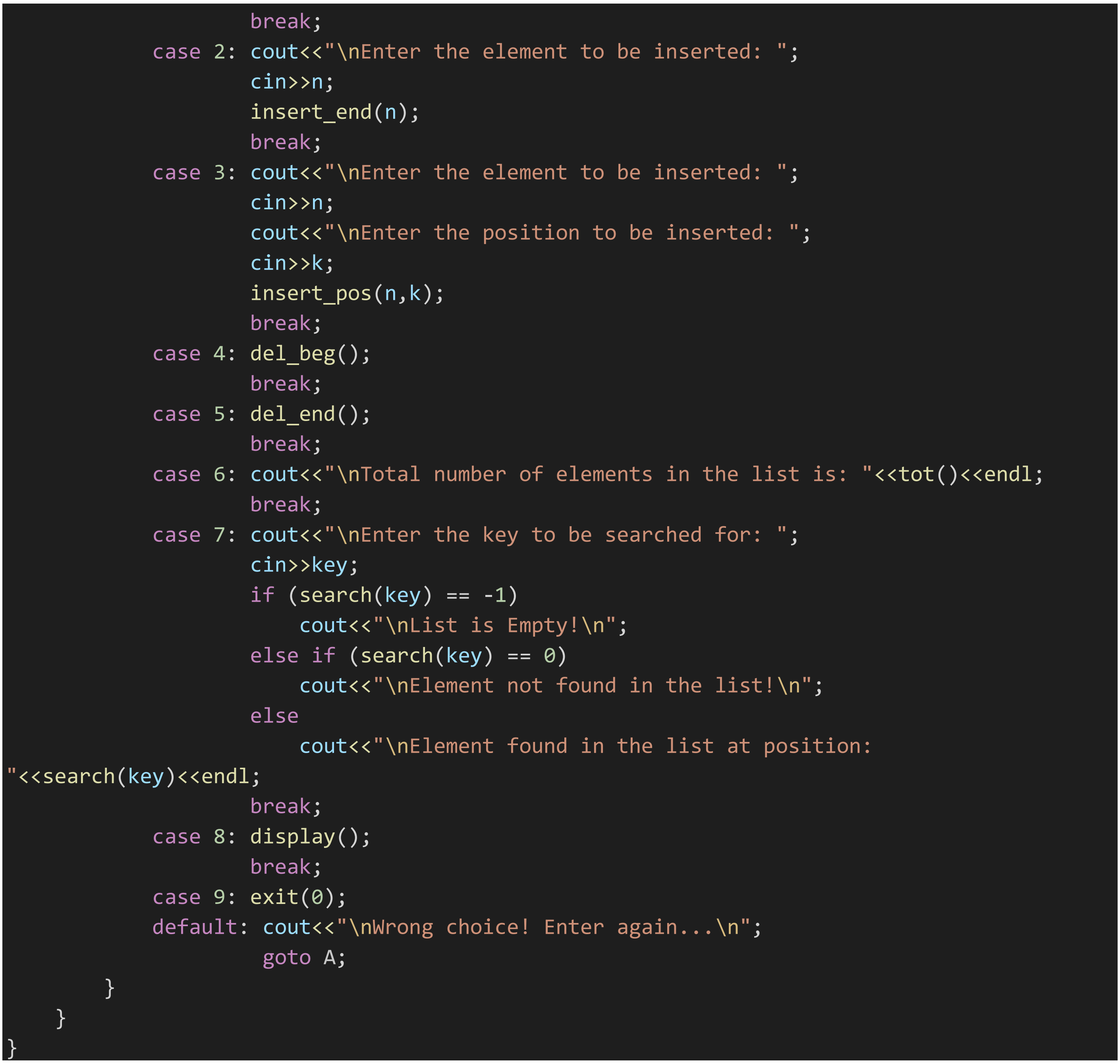
```
#include <iostream>
using namespace std;

struct list
{
    int info;
    struct list *next;
};
struct list *ptr, *temp, *p, *start=NULL, *rear=NULL;

void new_node (int n)
{
    ptr = (struct list *) malloc (sizeof(struct list));
    if (ptr==NULL)
    {
        cout<<"\nMemory could not be allocated!\n";
        return;
    }
    ptr->info = n;
    ptr->next = NULL;
}

int tot ()
{
    int c=0;
    if (start==NULL)
        return c;
    p=start;
    while (p != NULL)
    {
        p = p->next;
        c++;
    }
    return c;
}

void display ()
{
    if (tot()==0)
    {
        cout<<"\nList is empty, nothing to display!\n";
        return;
    }
    p=start->next;
    cout<<endl<<"List items: ";
    cout<<endl<<start->info;
    while (p!=NULL)
    {
        cout<<" -> "<<p->info;
        p=p->next;
    }
    cout<<" -> NULL"<<endl;
}

void insert_beg (int n)
{
    new_node(n);
    if (tot()==0)
    {
        start = ptr;
        rear = ptr;
    }
    else
    {
        // storing the previous first node's address to the next of the newly inserted node
        temp = start;
        start = ptr;
        ptr->next = temp;
    }
    display();
}

void insert_end (int n)
{
    new_node(n);
    if (tot()==0)
    {
        start = ptr;
        rear = ptr;
        return ;
    }
    rear->next = ptr;
    rear = ptr;
    display();
}

void insert_pos (int n, int k)
{
    if (k==1)
    {
        insert_beg(n);
        return;
    }
    else if (k > tot())
    {
```



```
        insert_end(n);
        return;
    }

    new_node(n);
    int c = 1;
    if (tot()==0)
    {
        cout<<"\nList is empty, inserting at first position.\n";
        start = ptr;
        rear = ptr;
    }
    else
    {
        p = start;
        while (c < (k-1))
        {
            p=p->next;
            c++;
        }
        temp = p->next;
        p->next = ptr;
        ptr->next = temp;
    }
    display();
}

void del_beg ()
{
    if (tot()==0)
    {
        cout<<"\nList is empty, cannot delete!\n";
        return;
    }
    if (tot()==1)
    {
        start = NULL;
        rear = NULL;
        display();
        return ;
    }
    else
    {
        ptr = start;
        start = start->next;
        delete ptr;
    }
    display();
}

void del_end()
{
    if (tot()==0)
    {
```

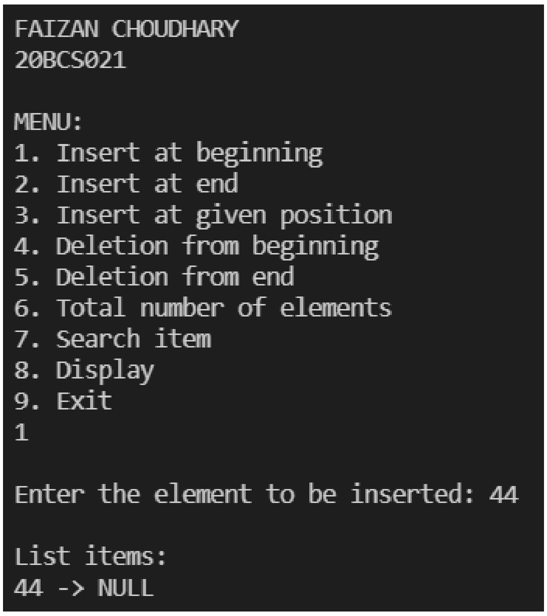


```
        cout<<"\nList is empty, cannot delete!\n";
        return;
    }
    if (tot()==1)
    {
        start = NULL;
        rear = NULL;
        display();
        return ;
    }
    temp = start;
    while (temp->next != rear)
        temp = temp->next;
    ptr = rear;
    temp->next = NULL;
    rear = temp;
    delete ptr;

    display();
}

int search (int n)
{
    int pos=0;
    if (tot()==0)
        return -1;
    p=start;
    while (p != NULL)
    {
        pos++;
        if (p->info == n)
            return pos;
        p = p->next;
    }
    return 0;
}

int main()
{
    cout<<"\nFAIZAN CHOUDHARY\n20BCS021\n";

    int ch,n,k,key;
    while (true)
    {
        A:
        cout<<"\nMENU:\n1. Insert at beginning\n2. Insert at end\n3. Insert at given
position\n4. Deletion from beginning\n5. Deletion from end\n6. Total number of
elements\n7. Search item\n8. Display\n9. Exit\n";
        cin>>ch;
        switch (ch)
        {
            case 1: cout<<"\nEnter the element to be inserted: ";
                    cin>>n;
                    insert_beg(n);
```

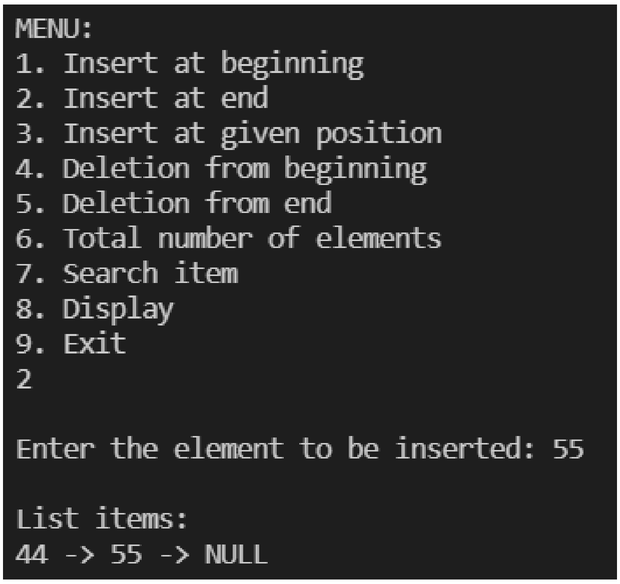


```
                    break;
            case 2: cout<<"\nEnter the element to be inserted: ";
                    cin>>n;
                    insert_end(n);
                    break;
            case 3: cout<<"\nEnter the element to be inserted: ";
                    cin>>n;
                    cout<<"\nEnter the position to be inserted: ";
                    cin>>k;
                    insert_pos(n,k);
                    break;
            case 4: del_beg();
                    break;
            case 5: del_end();
                    break;
            case 6: cout<<"\nTotal number of elements in the list is: "<<tot()<<endl;
                    break;
            case 7: cout<<"\nEnter the key to be searched for: ";
                    cin>>key;
                    if (search(key) == -1)
                        cout<<"\nList is Empty!\n";
                    else if (search(key) == 0)
                        cout<<"\nElement not found in the list!\n";
                    else
                        cout<<"\nElement found in the list at position: 
"<<search(key)<<endl;
                    break;
            case 8: display();
                    break;
            case 9: exit(0);
            default: cout<<"\nWrong choice! Enter again...\n";
                     goto A;
        }
    }
}
```

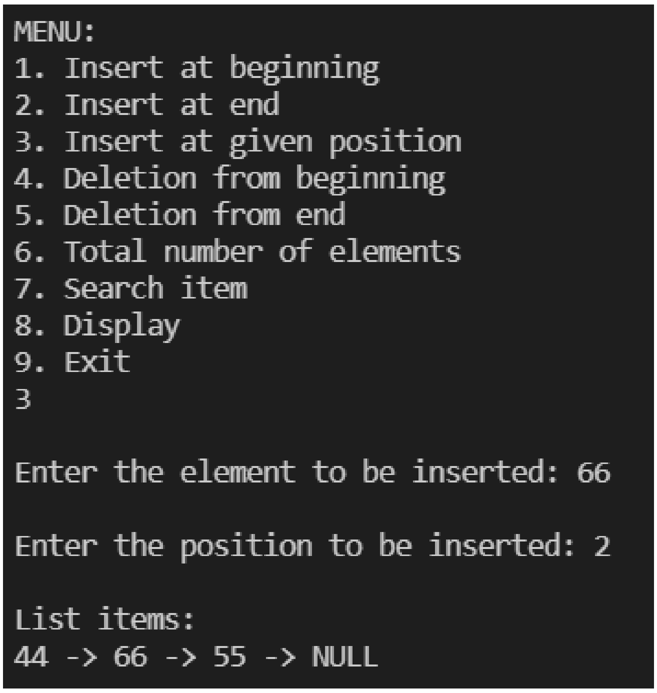
OUTPUT:



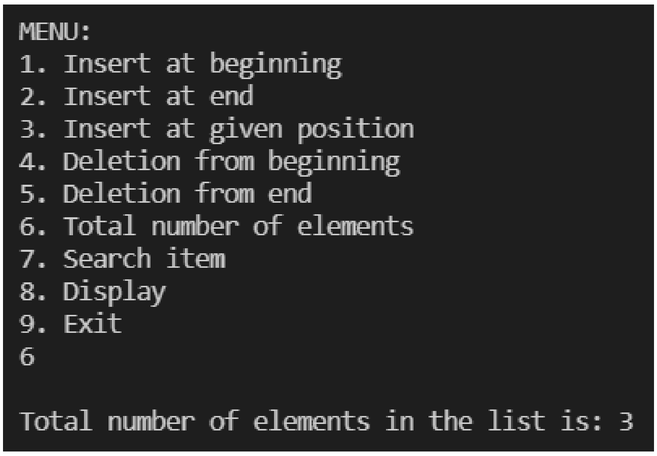
```
FAIZAN CHOUDHARY
20BCS021

MENU:
1. Insert at beginning
2. Insert at end
3. Insert at given position
4. Deletion from beginning
5. Deletion from end
6. Total number of elements
7. Search item
8. Display
9. Exit
1

Enter the element to be inserted: 44

List items:
44 -> NULL
```



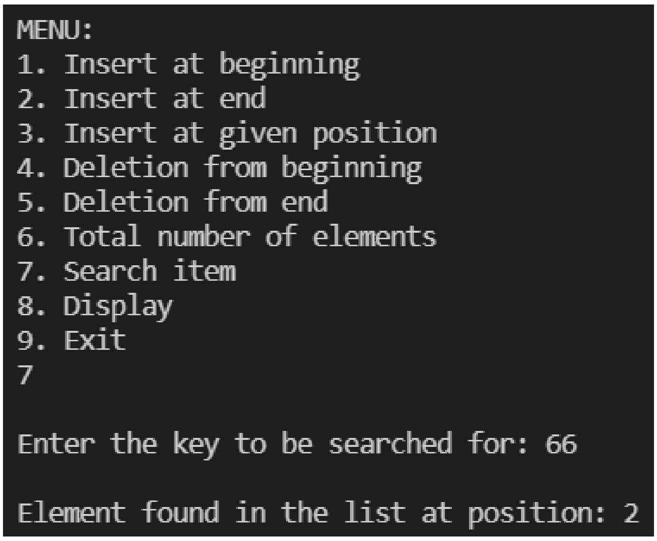
```
MENU:
1. Insert at beginning
2. Insert at end
3. Insert at given position
4. Deletion from beginning
5. Deletion from end
6. Total number of elements
7. Search item
8. Display
9. Exit
2

Enter the element to be inserted: 55

List items:
44 -> 55 -> NULL
```



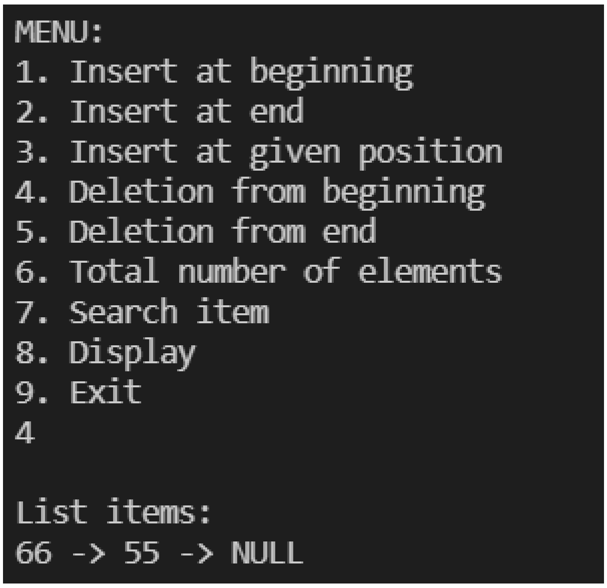
```
MENU:
1. Insert at beginning
2. Insert at end
3. Insert at given position
4. Deletion from beginning
5. Deletion from end
6. Total number of elements
7. Search item
8. Display
9. Exit
3

Enter the element to be inserted: 66

Enter the position to be inserted: 2

List items:
44 -> 66 -> 55 -> NULL
```



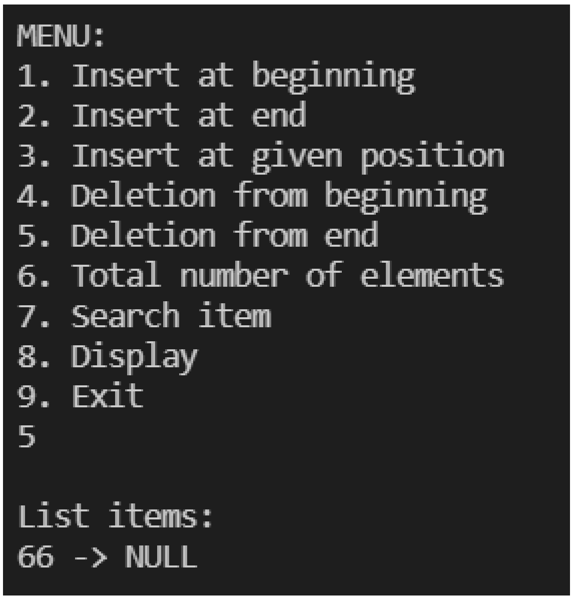
```
MENU:
1. Insert at beginning
2. Insert at end
3. Insert at given position
4. Deletion from beginning
5. Deletion from end
6. Total number of elements
7. Search item
8. Display
9. Exit
6

Total number of elements in the list is: 3
```



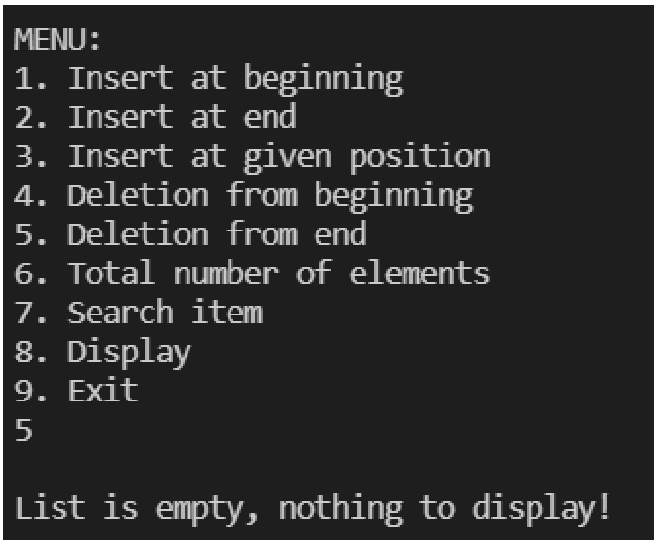
```
MENU:
1. Insert at beginning
2. Insert at end
3. Insert at given position
4. Deletion from beginning
5. Deletion from end
6. Total number of elements
7. Search item
8. Display
9. Exit
7

Enter the key to be searched for: 66

Element found in the list at position: 2
```



```
MENU:
1. Insert at beginning
2. Insert at end
3. Insert at given position
4. Deletion from beginning
5. Deletion from end
6. Total number of elements
7. Search item
8. Display
9. Exit
4

List items:
66 -> 55 -> NULL
```



```
MENU:
1. Insert at beginning
2. Insert at end
3. Insert at given position
4. Deletion from beginning
5. Deletion from end
6. Total number of elements
7. Search item
8. Display
9. Exit
5

List items:
66 -> NULL
```



```
MENU:
1. Insert at beginning
2. Insert at end
3. Insert at given position
4. Deletion from beginning
5. Deletion from end
6. Total number of elements
7. Search item
8. Display
9. Exit
5

List is empty, nothing to display!
```