Data Structure Lab

CEN-391

Practical Exam

Code :-

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

struct LinkedList
{
    int data;
    LinkedList *next;
    LinkedList *prev;
};
```

```
LinkedList *Create_NewNode()
{
    LinkedList *newnode = (LinkedList
*)malloc(sizeof(LinkedList));
    cout << "Enter The Element : ";
    cin >> newnode->data;
    newnode->next = nullptr;
    newnode->prev = nullptr;
    return newnode;
}

void Display(LinkedList *Head, int size)
{
    cout << "Display...\n";
    if (size == 0)
    {
        cout << "Linked List Is Empty!\n";
        return;
    }
    cout << "|Head|";
    while (Head)
    {
        cout << "--|" << Head->data << "|";
        Head = Head->next;
    }
    cout << "--|Tail|\n";
}

void Insert_At_End(LinkedList *&Head, LinkedList
*&Tail, int &size)
{
    cout << "Insert At End Operation Is Selected...
\n";
```

```
    LinkedList *newnode = Create_NewNode();
    if (size == 0)
    {
        size++;
        Head = newnode;
        Tail = newnode;
        Display(Head, size);
        return;
    }
    if (newnode == nullptr)
    {
        cout << "Memory Not Assigned!\n";
        return;
    }
    size++;
    Tail->next = newnode;
    newnode->prev = Tail;
    Tail = Tail->next;
    Display(Head, size);
}

void Delete_At_End(LinkedList *&Head, LinkedList
*&Tail, int &size)
{
    cout << "Delete At End Operation Is Selected...
\n";
    if (size == 0)
    {
        cout << "Linked List Underflow!\n";
        return;
    }
    size--;
    LinkedList *todelete = Tail;
    Tail = Tail->prev;
```

```
        Tail->next = nullptr;
        cout << todelete->data << "\n";
        delete todelete;
        if (size == 0)
        {
            Head == nullptr;
            Tail == nullptr;
        }
        Display(Head, size);
    }

    void Minimum_Element_In_Linked_List(LinkedList *Head,
    int size)
    {
        cout << "Minimum Element In Linked List Operation
    Is Selected... \n";
        if (size == 0)
        {
            cout << "Empty List!\n";
            return;
        }
        int Min = 1e9;
        LinkedList *curr = Head;
        while (curr != nullptr)
        {
            if (Min > curr->data)
                Min = curr->data;
            curr = curr->next;
        }
        cout << "Minimum Element : " << Min << "\n";
        Display(Head, size);
    }

    void Bars()
```

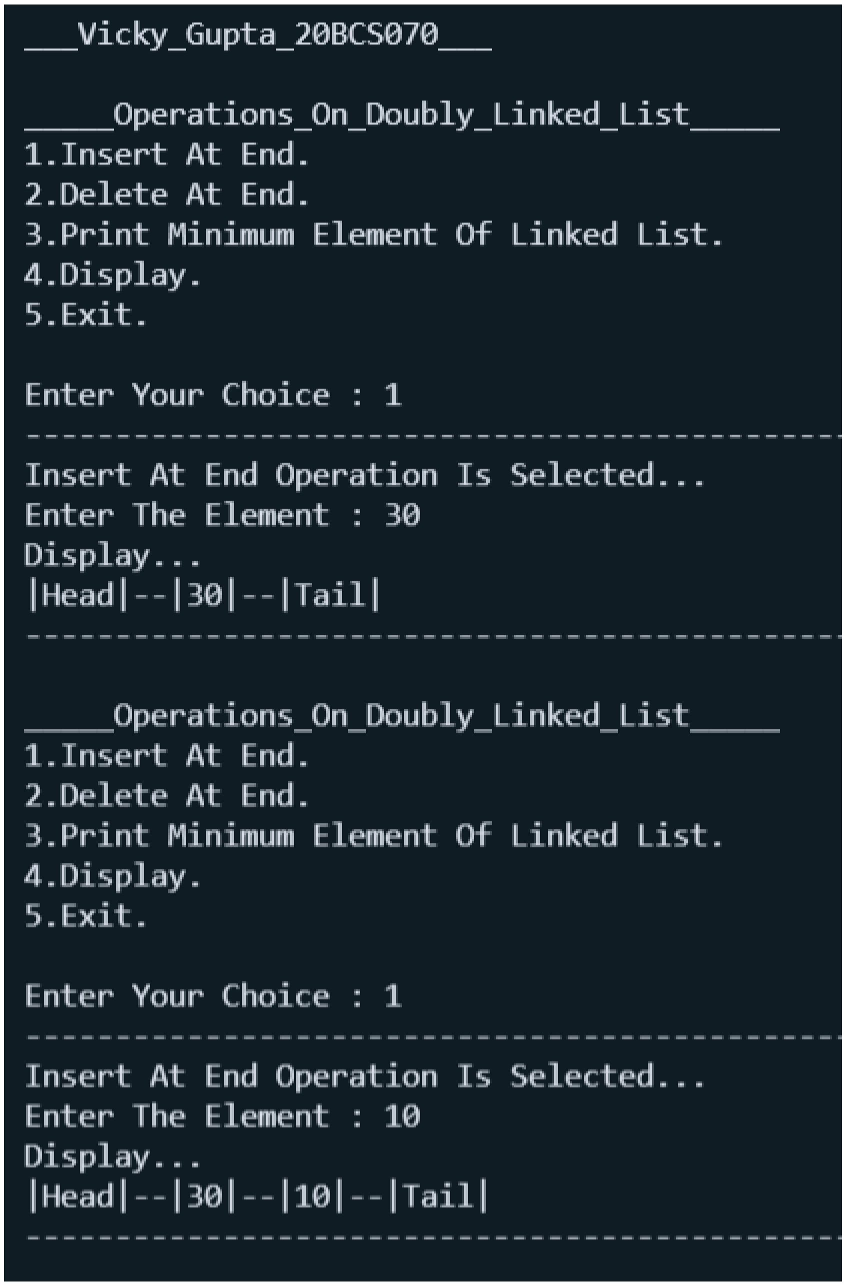
```
{
    cout << "----------------------------------------
-----------------------\n";
}

bool Options(LinkedList *&Head, LinkedList *&Tail,
int &size)
{
    int opt;
    cin >> opt;
    Bars();
    switch (opt)
    {
    case 1:
        Insert_At_End(Head, Tail, size);
        break;
    case 2:
        Delete_At_End(Head, Tail, size);
        break;
    case 3:
        Minimum_Element_In_Linked_List(Head, size);
        break;
    case 4:
        Display(Head, size);
        break;
    case 5:
        return 0;
        break;
    default:
        cout << "Invalid Input!\nTry Again!\n\n";
    }
    Bars();
    return 1;
}
```

```
void Menu()
{
    cout <<
"\n_____Operations_On_Doubly_Linked_List_____ \n";
    cout << "1.Insert At End. \n";
    cout << "2.Delete At End. \n";
    cout << "3.Print Minimum Element Of Linked List.
\n";
    cout << "4.Display. \n";
    cout << "5.Exit. \n";
    cout << "\nEnter Your Choice : ";
}

int main()
{
    system("cls");
    cout << "___Vicky_Gupta_20BCS070___\n";
    LinkedList *Head = nullptr, *Tail = nullptr;
    int size = 0;
    while (true)
    {
        Menu();
        if (!Options(Head, Tail, size))
            break;
    }
    cout << "Exiting...\n";
    Bars();
    return 0;
}
```

Output :-



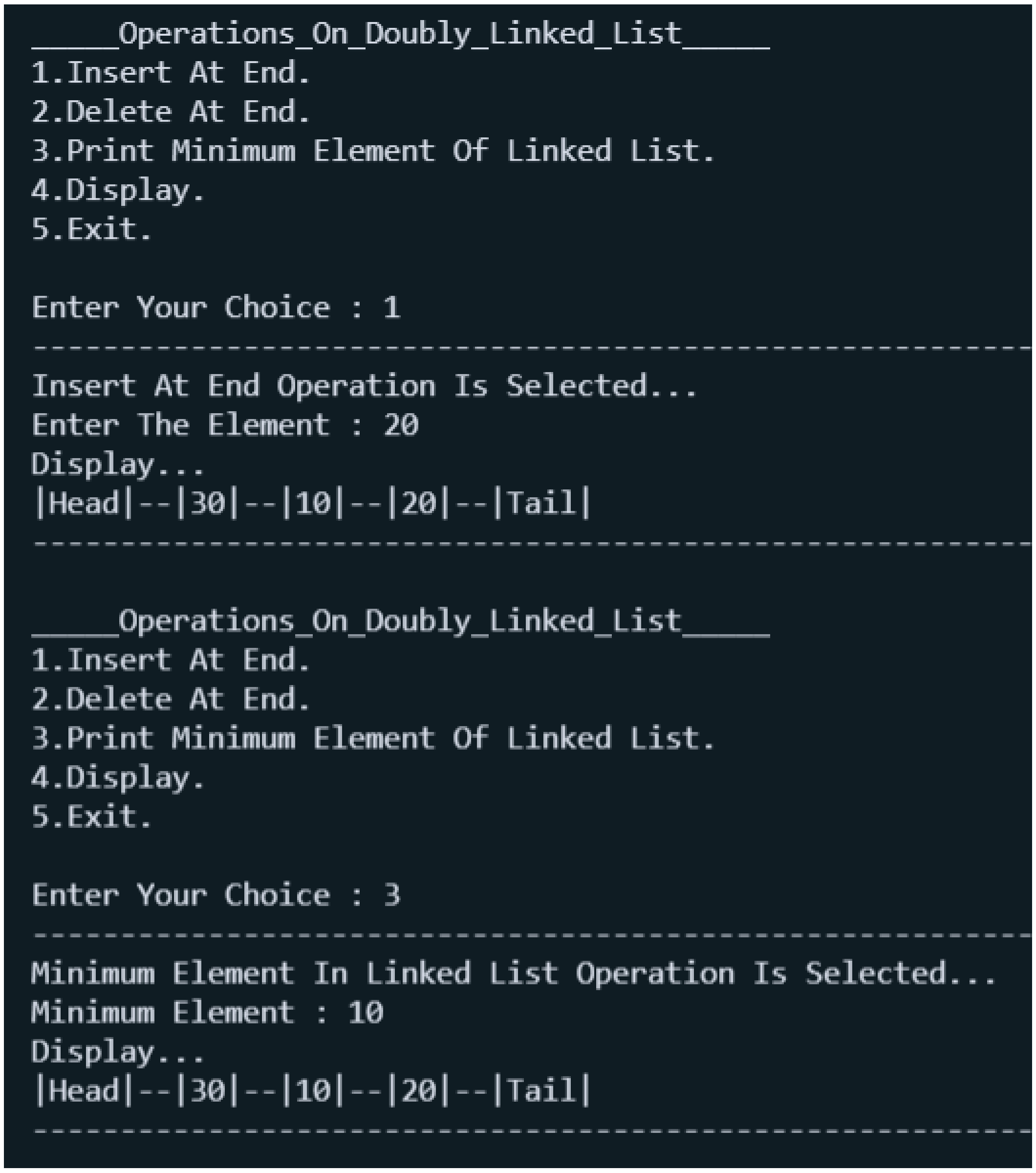
```
___Vicky_Gupta_20BCS070___

_____Operations_On_Doubly_Linked_List_____
1.Insert At End.
2.Delete At End.
3.Print Minimum Element Of Linked List.
4.Display.
5.Exit.

Enter Your Choice : 1
---------------------------------------------
Insert At End Operation Is Selected...
Enter The Element : 30
Display...
|Head|--|30|--|Tail|
---------------------------------------------

_____Operations_On_Doubly_Linked_List_____
1.Insert At End.
2.Delete At End.
3.Print Minimum Element Of Linked List.
4.Display.
5.Exit.

Enter Your Choice : 1
---------------------------------------------
Insert At End Operation Is Selected...
Enter The Element : 10
Display...
|Head|--|30|--|10|--|Tail|
---------------------------------------------
```

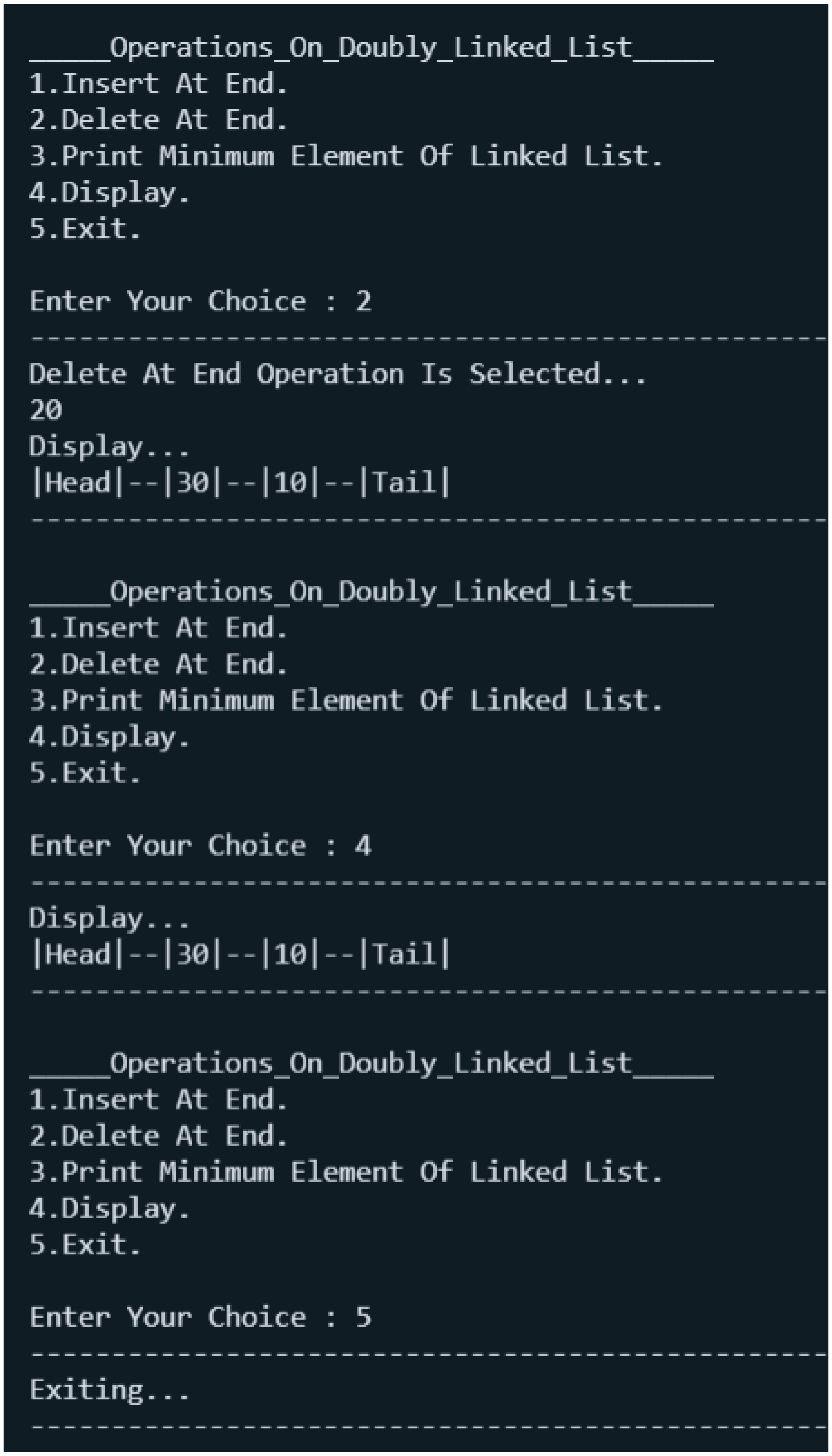


```
_____Operations_On_Doubly_Linked_List_____
1.Insert At End.
2.Delete At End.
3.Print Minimum Element Of Linked List.
4.Display.
5.Exit.

Enter Your Choice : 1
-----------------------------------------------------------
Insert At End Operation Is Selected...
Enter The Element : 20
Display...
|Head|--|30|--|10|--|20|--|Tail|
-----------------------------------------------------------

_____Operations_On_Doubly_Linked_List_____
1.Insert At End.
2.Delete At End.
3.Print Minimum Element Of Linked List.
4.Display.
5.Exit.

Enter Your Choice : 3
-----------------------------------------------------------
Minimum Element In Linked List Operation Is Selected...
Minimum Element : 10
Display...
|Head|--|30|--|10|--|20|--|Tail|
-----------------------------------------------------------
```



```
_____Operations_On_Doubly_Linked_List_____
1.Insert At End.
2.Delete At End.
3.Print Minimum Element Of Linked List.
4.Display.
5.Exit.

Enter Your Choice : 2
--------------------------------------------------
Delete At End Operation Is Selected...
20
Display...
|Head|--|30|--|10|--|Tail|
--------------------------------------------------

_____Operations_On_Doubly_Linked_List_____
1.Insert At End.
2.Delete At End.
3.Print Minimum Element Of Linked List.
4.Display.
5.Exit.

Enter Your Choice : 4
--------------------------------------------------
Display...
|Head|--|30|--|10|--|Tail|
--------------------------------------------------

_____Operations_On_Doubly_Linked_List_____
1.Insert At End.
2.Delete At End.
3.Print Minimum Element Of Linked List.
4.Display.
5.Exit.

Enter Your Choice : 5
--------------------------------------------------
Exiting...
--------------------------------------------------
```