PRN: 2020BTEIT00041

CODE:

```
/*
    Name: Om Vivek Gharge
    PRN : 2020BTEIT00041
    Batch: S3

    Program: Fibonacci series using recursion
*/

#include<iostream>
using namespace std;

int Fibonacci(int n)
{
    if((n == 1) || (n==0))
    {
        return (n);
    }
    else
    {
        return (Fibonacci(n-1)+Fibonacci(n-2));
    }
}

int main()
{

    int n;
    cout<<"Enter the no. of terms to be displayed: ";
    cin>>n;
    for(int i=0;i<n;i++)
    {

    cout<<" "<<Fibonacci(i);

    }

    return 0;
}
```

OUTPUT:

```
Enter the no. of terms to be displayed: 5
 0 1 1 2 3
```

CODE:

```cpp
/*
    Name: Om Vivek Gharge
    PRN : 2020BTEIT00041
    Batch: S3

    Program: Doubly Linked List (insert front , delete rear , display)
*/

#include<iostream>
using namespace std;

class Node{
    public:

    int data;
    Node* next;
    Node* prev;

    Node(int data)
    {
        this->data = data;
        this->next = NULL;
        this->prev = NULL;
    }

};

class Head{
    public:

    int count;
    Node* first;
    Node* last;

    Head(int count, Node* f, Node* l){
        this->count = count;
        this->first = f;
        this->last = l;
    }
};

void insertAtHead(Head* h, int data){

    Node* new_node = new Node(data);

    if(h->count == 0){
        h->first = new_node;
        h->last = new_node;
        h->count++;
    }
    else{
        new_node->next = h->first;
        h->first->prev = new_node;

        h->first = new_node;
        h->count++;
    }
}

void printListHead(Head* h,int s){
    Node* p = h->first;

    for(int i=0; i<s; i++){
        cout<<p->data<<" ";
        p = p->next;
    }
}


void delNodeAt(Head* h, int location){

    if(h->count==0){
        cout<<"List is empty, can't delete."<<"\n";
        return;
    }

    Node* p = h->first;

```

```
    //  To Del head
    if(location==0){
        h->first = p->next;
        p->next = NULL;
        free(p);
        h->count--;

        return;
    }

    Node* q;
    for(int i=1; i<h->count; i++){

        if(i==location){
            q->next = p->next;
            p->next = NULL;
            free(p);
            h->count--;

            return;
        }

        q = p;
        p = p->next;
    }

}

int main()
{

    Head* h = new Head(0, NULL, NULL);
    int opt, data, index;
    int choice;
    int size=0;
    while(1)
    {
        cout<<"\nMENU:\n 1. Insert front\n 2. Delete rear \n 3. Display \n4. Exit\n";
        cout<<"Choose : ";
            cin>>choice;
            switch (choice)
            {
            case 1:
                 cout<<"Enter data to add: ";
                        cin>>data;
                        size++;
                        cout<<"Adding data...\n";
                        insertAtHead(h, data);
                        cout<<"\n";
                break;

            case 2:
                    if(size!=0)
                    {
                        cout<<"Deleting data...\n";
                        delNodeAt(h,size);
                        cout<<"\n";
                        size--;
                    }
                    else
                    {
                        cout<<"Cannot delete, Doubly Linked List is empty"<<endl;
                    }
                    break;
            case 3:
            if(size>0)
            {
                        cout<<"Displaying the LinkedList from Head: ";
                        printListHead(h,size);
                        cout<<"\n";
            }
            else
            {
                        cout<<"Doubly Linked List is Empty.."<<endl;
            }
                        break;
```

```
            case 4:
                return 0;
                break;
            default:
                break;
            }

        }

    return 0;
    }
```

OUTPUT:

```
MENU:
 1. Insert front
 2. Delete rear
 3. Display
4. Exit
Choose : 1
Enter data to add: 10
Adding data...


MENU:
 1. Insert front
 2. Delete rear
 3. Display
4. Exit
Choose : 1
Enter data to add: 20
Adding data...


MENU:
 1. Insert front
 2. Delete rear
 3. Display
4. Exit
Choose : 1
Enter data to add: 30
Adding data...


MENU:
 1. Insert front
 2. Delete rear
 3. Display
4. Exit
Choose : 3
Displaying the LinkedList from Head: 30 20 10
```

```
MENU:
 1. Insert front
 2. Delete rear
 3. Display
4. Exit
Choose : 2
Deleting data...


MENU:
 1. Insert front
 2. Delete rear
 3. Display
4. Exit
Choose : 3
Displaying the LinkedList from Head: 30 20
```

```
MENU:
 1. Insert front
 2. Delete rear
 3. Display
4. Exit
Choose : 4
PS F:\Assignments\DSA\Asst_17_11_21>
```