

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:

```
1  /*
2   Name: Om Vivek Gharge
3   PRN : 2020BTEIT00041
4   Batch: S3
5
6   Program: Doubly Linked List (insert front , delete rear , display)
7  */
8
9  #include<iostream>
10 using namespace std;
11
12 class Node{
13 public:
14
15     int data;
16     Node* next;
17     Node* prev;
18
19     Node(int data)
20     {
21         this->data = data;
22         this->next = NULL;
23         this->prev = NULL;
24     }
25 };
26
27 class Head{
28 public:
29
30     int count;
31     Node* first;
32     Node* last;
33
34     Head(int count, Node* f, Node* l){
35         this->count = count;
36         this->first = f;
37         this->last = l;
38     }
39 }
```

```
40 };
41
42 void insertAtHead(Head* h, int data){
43
44     Node* new_node = new Node(data);
45
46     if(h->count == 0){
47         h->first = new_node;
48         h->last = new_node;
49         h->count++;
50     }
51     else{
52         new_node->next = h->first;
53         h->first->prev = new_node;
54
55         h->first = new_node;
56         h->count++;
57     }
58 }
59
60 void printListHead(Head* h,int s){
61     Node* p = h->first;
62
63     for(int i=0; i<s; i++){
64         cout<<p->data<<" ";
65         p = p->next;
66     }
67 }
68
69
70 void delNodeAt(Head* h, int location){
71
72     if(h->count==0){
73         cout<<"List is empty, can't delete."<<"\n";
74         return;
75     }
76
77     Node* p = h->first;
78 }
```

```

79 // To Del head
80 if(location==0){
81     h->first = p->next;
82     p->next = NULL;
83     free(p);
84     h->count--;
85
86     return;
87 }
88
89 Node* q;
90 for(int i=1; i<h->count; i++){
91
92     if(i==location){
93         q->next = p->next;
94         p->next = NULL;
95         free(p);
96         h->count--;
97
98         return;
99     }
100
101     q = p;
102     p = p->next;
103 }
104
105 }
106
107 int main()
108 {
109
110     Head* h = new Head(0, NULL, NULL);
111     int opt, data, index;
112     int choice;
113     int size=0;
114     while(1)
115     {
116         cout<<"\nMENU:\n 1. Insert front\n 2. Delete rear \n 3. Display \n 4. Exit\n";
117         cout<<"Choose : ";

```

```

118         cin>>choice;
119         switch (choice)
120         {
121             case 1:
122                 cout<<"Enter data to add: ";
123                 cin>>data;
124                 size++;
125                 cout<<"Adding data...\n";
126                 insertAtHead(h, data);
127                 cout<<"\n";
128                 break;
129
130             case 2:
131                 if(size!=0)
132                 {
133                     cout<<"Deleting data...\n";
134                     delNodeAt(h,size);
135                     cout<<"\n";
136                     size--;
137                 }
138                 else
139                 {
140                     cout<<"Cannot delete, Doubly Linked List is empty"<<endl;
141                 }
142                 break;
143
144             case 3:
145                 if(size>0)
146                 {
147                     cout<<"Displaying the LinkedList from Head: ";
148                     printListHead(h,size);
149                     cout<<"\n";
150                 }
151                 else
152                 {
153                     cout<<"Doubly Linked List is Empty.."<<endl;
154                 }
155                 break;

```

```

155         case 4:
156             return 0;
157             break;
158         default:
159             break;
160     }
161 }
162 }
163
164 return 0;
165 }

```

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>

```