6. Implement stacks using Linked list

CODE :

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
using namespace std;
struct Node
{
   int data;
   struct Node *next;
} *top = NULL;

void push(int x)
{
   struct Node *t;
   t = new Node();
   if (t == NULL)
        cout << "stack is full\n";
   else
   {
        t->data = x;
        t->next = top;
        top = t;
   }
}
int pop()
{
   struct Node *t;
   int x = -1;
   if (top == NULL)
        cout << "Stack is Empty\n";
   else
   {
        t = top;
```

```
        top = top->next;
        x = t->data;
        free(t);
    }
    return x;
}
void isEmpty()
{
    if (top == NULL)
    {
        cout << "Stack is Empty\n";
    }
    else
    {
        cout << "Stack is not Empty \n";
    }
}
void peek()
{
    if (top == NULL)
        cout << "Stack is Empty\n";

    else
    {
        cout << top->data << endl;
    }
}
void Display()
{
    struct Node *p;
    p = top;
    while (p != NULL)
    {
        cout << p->data;
        cout << endl;
        p = p->next;
    }
    cout << "\n";
}

int main()
{
```

```
   int ch, val;
   cout << "1)Push\n";
   cout << "2)Pop\n";
   cout << "3)Is Empty\n";
   cout << "4)Peek\n";
       cout << "5)Display\n";
           cout << "6)Exit\n";
   do
   {
       cout << "Enter choice : " << endl;
       cin >> ch;
       switch (ch)
       {
       case 1:
           cout << "Input for insertion: " << endl;
           cin >> val;
           push(val);
           break;
       case 2:
           pop();
           break;
       case 3:
           isEmpty();
           break;
       case 4:
           peek();
           break;
       case 5:
           Display();
           break;
       default:
           cout << "Incorrect!\n";
       }
   } while (ch != 6);

    return 0;
}
```

OUTPUT :

```
/Users/amzamani/Desktop/sem3/DS/ds lab/assgn06/"StackLinkedL
1)Push
2)Pop
3)Is Empty
4)Peek
5)Display
6)Exit
Enter choice :
1
Input for insertion:
10
Enter choice :
1
Input for insertion:
20
Enter choice :
1
Input for insertion:
30
Enter choice :
5
30
20
10

Enter choice :
4
30
Enter choice :
3
Stack is not Empty
Enter choice :
5
30
20
10

Enter choice :
6
```