

```

#include <iostream>
using namespace std;

template <class T>
class StackSinglelink
{
private:

    template <class S>
    struct Node
    {
        int value;
        Node *next;

        Node(S value)
        {
            this->value = value;
            next = nullptr;
        }
    };

    Node<T> *head;
    Node<T> *tail;

public:
    StackSinglelink()
    {
        head=nullptr;
        tail=nullptr;
    }

    void display()
    {
        Node<T> *current;
        for( current=head; current; current=current->next)
        {
            cout<<current->value<<'\\t';
        }
        cout<<endl;
    }

    void push(T num)
    {
        Node<T> *newnode=new Node<T>(num);
        if(nullptr==head)
            tail=newnode;
        else
            newnode->next=head;
        head=newnode;
    }

    T pop()
    {
        T num;
        Node<T> *nodeDelete = head;
        if(head != tail)
        {
            head = head->next;
        }
    }

```

```

        num = nodeDelete->value;
        delete nodeDelete;
        return num;
    }
    if(nodeDelete)
    {
        num = nodeDelete->value;
        delete nodeDelete;
        head = nullptr;
        tail = nullptr;
        return num;
    }
    return 0;
}

};

int main()
{
    StackSinglelink<int> a;
    int option, num, returnValue;

    while(true)
    {
        cout<<endl<<"enter the options"<<endl<<"1.to enter data"<<endl<<"2.to
display stack"<<endl<<"3.to remove last entry"<<endl<<"4.to exit"<<endl;
        cout<<"enter your choice"<<endl;
        cin>>option;

        switch (option)
        {
            case 1:int value;
                cout<<endl<<"enter the value";
                cin>>value;
                a.push(value);
                break;

            case 2:a.display();
                break;

            case 3:int pop;
                pop =a.pop();
                cout<<pop<<"is removed"<<endl;
                break;

            case 4:cout<<"exited"<<endl;
                return 0;

            default:cout<<endl<<"try again"<<endl;
        }
    }
    return 0;
}

```