Week 6: Array

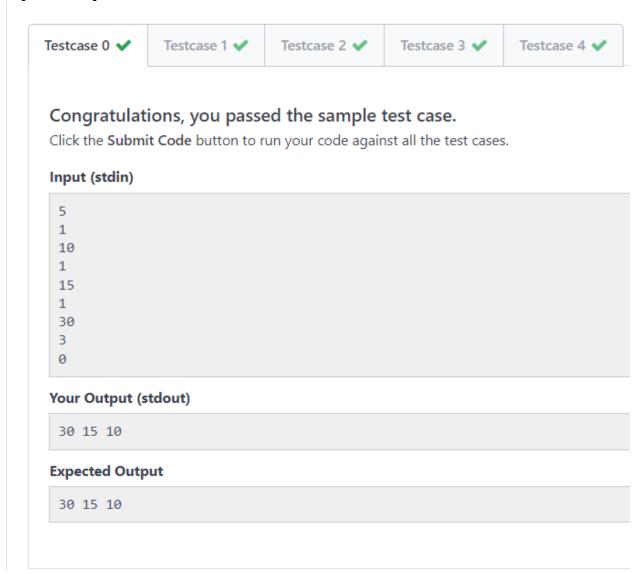
Aim: Implement basic Operations(Push(), pop() and display()) of stack using array.

Program:

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
class stack{
    int *arr;
    int top;
    int size;
public:
    stack()
        top = -1;
       size = 1000;
        arr = new int[1000];
    stack(int x)
        top = -1;
        size =x;
        arr = new int[x];
    void push(int x)
        top++;
        if(top >= size)
            cout<<"overflow";</pre>
            return;
        arr[top] = x;
    void pop()
        if(top == -1)
            cout<<"underflow"<<endl;</pre>
```

```
top--;
    void display()
        if(top <0)
            return;
        for(int i=top;i>=0;i--)
            cout<<arr[i]<<" ";</pre>
        cout<<" ";
};
int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT
*/ int t;
    cin>>t;
    stack s(t);
    cin>>t;
    while(t!= 0)
        if(t == 1)
            cin>>n;
            s.push(n);
        else if(t == 2)
            s.pop();
        else if(t == 3)
            s.display();
        cin>>t;
    return 0;
```

Input & Output:



Conclusion: From the above program I have learned to implement the Stack using array.

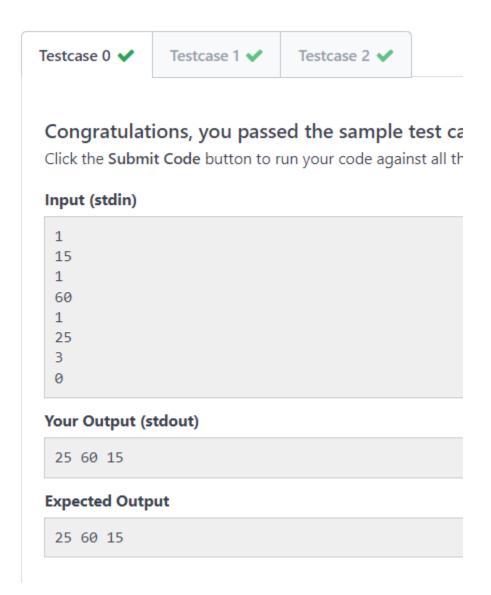
Aim: Implement basic Operations(Push(), pop() and display()) of stack using linked list.

Program:

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;
class Node
public:
    int val;
    Node *next;
    Node()
        val = 0;
        next = NULL;
    Node(int x)
        val = x;
        next = NULL;
void insert(Node *&head, int x)
    Node *temp = new Node(x);
    if (head == NULL)
        head = temp;
        return;
    temp->next = head;
    head = temp;
class stack
    Node *head = NULL;
public:
    void push(int x)
        insert(head, x);
```

```
void pop()
        if (head == NULL)
            cout << "underflow" << endl;</pre>
            return;
        head = head->next;
    void display()
        Node *temp = head;
        while (temp != NULL)
             cout << temp->val << " ";</pre>
            temp = temp->next;
        cout << endl;</pre>
};
int main()
    stack s;
    int t;
    cin >> t;
    while (t != 0)
        if (t == 1)
            int x;
            cin >> x;
            s.push(x);
        else if (t == 2)
            s.pop();
        else if (t == 3)
            s.display();
        cin >> t;
    return 0;
```

Input & Output:



Conclusion: From the above program I have learned to implement the Stack using Linked List.