

```
#include<iostream>
#include<stdlib.h>
#include<conio.h>
using namespace std;
template<class T>class Stack_Array
{
    T *items;
    int top,size;
public:
    Stack_Array(int len)
    {
        items=new T[len];
        top=-1;
        size=len;
    }
    bool isempty()
    {
        return( top== -1 );
    }
}
```

```
}  
  
bool isFull()  
{  
    if(top==size-1)  
        return true;  
    else  
        return false;  
}  
  
void push(T val)  
{  
    if(isFull())  
        cout<<"Overflow Condition ";  
    else  
    {  
        top++;  
        items[top]=val;  
    }  
}
```

```
T pop()
{
    if(isempty())
    {
        cout<<"Underflow Condition ";
        return -1;
    }
    else
    {
        T val=items[top];
        top--;
        return val;
    }
}

void Clear()
{
    top=-1;
}
```

```
void Display()
{
    cout<<"Stack : ";
    if(isempty())
        cout<<"Empty ";
    else
    {
        int k=top;
        while(k!=-1)
        {
            cout<<items[k]<<" ";
            k=k-1;
        }
    }
}

void menu()
{
    cout<<"MENU";
```

```
    cout<<"\n1.PUSH. ";
    cout<<"\n2.POP. ";
    cout<<"\n3.Display. ";
    cout<<"\n4.Clear. ";
    cout<<"\n5.Exit. ";
    choice();
}

void choice()
{
    T val,r;
    int ch;
    char c='Y';
    cout<<"\nEnter your choice : ";
    cin>>ch;
    switch(ch)
    {
        case 1: cout<<"Enter data : ";
                cin>>val;
```

```
        push(val);  
        break;  
    case 2: r=pop();  
        cout<<"Value Deleted : "<<r;  
        break;  
    case 3: Display();  
        break;  
    case 4: Clear();  
        break;  
    case 5: exit(0);  
    default: cout<<"Wrong input!! ";  
}  
cout<<"\nDo you want to  
continue(Y/N) : ";  
cin>>c;  
if(c=='Y' || c=='y')  
    choice();  
else
```

```
        cout<<"Press any key to exit ";  
    }  
};  
  
int main()  
{  
    int s;  
    cout<<"Enter the size of stack : ";  
    cin>>s;  
    Stack_Array<int> S(s);  
    Stack_Array<float> S1(s);  
    S.menu();  
    S1.menu();  
    getch();  
    return 0;  
}
```


