Stack implementation using Array

```
Enter the number of elements in the stack:  5

        ********* Stack operations using Array *********
        ----------------------------------------------

        Choose one from the following options...

        1.Push
        2.Pop
        3.Show
        4.Exit

Enter your choice: 1
Enter the value: 22

The Item is Pushed!

        Choose one from the following options...

        1.Push
        2.Pop
        3.Show
        4.Exit

Enter your choice:
```

```
// Array Based Stack implementation in C++
#include <iostream>
#include <stdlib.h>
using namespace std;

//public variable declaration
int stack[100],i,choice=0,n,top=-1;

//Function declaration
bool isempty();
bool isfull();
void push();
void pop();
void peek();
```

```
//main function definition
int main ()
{
   cout<<"Enter the number of elements in the stack:  ";
   cin>>n;
   cout<<"\n\t********* Stack operations using Array
*********";
    cout<<"\n\t----------------------------------------\n";
   while(choice != 4)
   {
      cout<<"\n\tChoose one from the following
options...\n";
      cout<<"\n\t1.Push\n\t2.Pop\n\t3.Show\n\t4.Exit";
      cout<<"\n\nEnter your choice: ";
      cin>>choice;
```

```
switch(choice)   {
        case 1:
        {
           push();
           break;
        }
        case 2:
        {
           pop();
           break;
        }
        case 3:
        {
           peek();
           break;
        }
```

```
case 4:
        {
           cout<<"\nExiting....";
           break;
        }
        default:
        {
           cout<<"\nPlease Enter valid
choice! ";
        }
      };  //end of switch statement
   }  //end of while loop
} //end of main function
```

```
//isempty function definition
bool isempty()
{
    if(top==-1)
        return true;
    else
        return false;
}
```

```
//isfull function definition
bool isfull()
{
    if(top==n-1)
        return true;
    else
        return false;
}
```

```
//push function definition
void push()
{
   int val;
   if (isfull())  {
         cout<<"\nOverflow! The Stack is Full";
    }
   else
   {
      cout<<"Enter the value: ";
      cin>>val;
      top = top +1;
      stack[top] = val;
         cout<<"\nThe Item is Pushed!";
   }
}
```

```
//pop function definition
void pop()
{
   if(isempty())
    {
        cout<<"\nUnderflow! The Stack is Empty!";
    }
   else
    {
        top = top -1;
        cout<<"\nItem Popped!";
    }
}
```

```
//Peek function definition
void peek()
{
   if(isempty())
   {
    cout<<"\nUnderflow! The Stack is Empty";
    }
    else
    {
        cout<<"\nList of Elements in the Stack:  ";
        for (i=top;i>=0;i--)
            cout<<stack[i]<<" ";
    }
}
```

Stack implementation using Linked List

```
        ********* Stack operations using Linked List *********
        ----------------------------------------------------

        Choose one from the following options...

        1.Push
        2.Pop
        3.Display
        4.Exit

Enter your choice: 1

Enter the data: 22

The Item is Pushed!

        Choose one from the following options...

        1.Push
        2.Pop
        3.Display
        4.Exit

Enter your choice:
```

```
//Linked List Based Stack implementation in C++
#include <iostream>
#include <stdlib.h>
using namespace std;

//Function declaration
void push();
void pop();
void peek();

//Node Creation
struct node {
    int val;
    struct node *next;
};
struct node *head = NULL;
```

```
//main function definition
int main ()
{
    int choice = 0;
   cout<<"\n\t********* Stack operations using Linked List *********";
    cout<<"\n\t----------------------------------------\n";
while(choice != 4)
   {
      cout<<"\n\tChoose one from the following options...\n";
      cout<<"\n\t1.Push\n\t2.Pop\n\t3.Show\n\t4.Exit";
      cout<<"\n\nEnter your choice: ";
      cin>>choice;
```

```
switch(choice)   {
        case 1:
        {
           push();
           break;
        }
        case 2:
        {
           pop();
           break;
        }
        case 3:
        {
           peek();
           break;
        }
```

```
case 4:
        {
           cout<<"\nExiting....";
           break;
        }
        default:
        {
           cout<<"\nPlease Enter valid
choice! ";
        }
      };  //end of switch statement
   }  //end of while loop
} //end of main function
```

```
//push function definition
void push ()
{
   int data;
   struct node *ptr = (struct node*)malloc(sizeof(struct node));
   if(ptr == NULL)
   {
      cout<<"\nSorry!, Not able to push the element";
   }
   else
   {
      cout<<"\nEnter the data: ";
      cin>>data;
```

```
if(head==NULL)
        {
            ptr->val = data;
            ptr -> next = NULL;
            head=ptr;
        }
        else
        {
            ptr->val = data;
            ptr->next = head;
            head=ptr;
        }
        cout<<"\nThe Item is Pushed! \n";
    }  //end of the first 'else' statement
} //end of push function
```

```
//pop function definition
void pop()  {
   int item;
   struct node *ptr;
   if(head==NULL) {
    cout<<"\nUnderflow! The Stack is Empty";
    }
   else
   {
      item = head->val;
      ptr = head;
      head = head->next;
      delete ptr;
      cout<<"\nItem popped\n";
   }
}
```

```
//Peek function definition
void peek()  {
   int i;
   struct node *ptr;
   ptr=head;
   if(ptr==NULL)  {
     cout<<"\nUnderflow! The Stack is Empty";
     }
   else
   {
      cout<<"\nList of Stack Elements: ";
      while(ptr!=NULL)
      {
         cout<<ptr->val<<" ";
         ptr = ptr->next;
      }
   }
}
```