

Pop, Push, isEmpty, IsFull:

Pseudocode:

Declare maxsize = 2

Declare stack [max.size]

Declare top = -1

Function isEmpty()

return (top == -1)

Function IsFull()

return (top == MaxSize - 1)

Function push(value)

If IsFull() then

Print ("Stack is full or Overflow")

Else

top = top + 1

stack [top] = value

print value + "pushed to stack."

Function pop (Value)

If isEmpty() then

Print "Stack is empty or Underflow"

return -1

Else

value = stack [top]

top = top - 1

Print value + "popped from stack."

return value

Function display

If isEmpty() then

print "Stack is Empty"

Else

Print "Stack elements are:"

for from i from top DownTo 0 DO

Print [stack][i]

End.

Program 1

```
#include <stdio.h>
#include <conio.h>
#define max 2

int top = -1, i, item, ch, S[10];
void push();
void pop();
void display();

int main()
{
    while(1)
    {
        printf("Perform operations on the stack:\n");
        printf("1: Push\n 2: Pop\n 3: Display\n 4. End");
        printf("Enter the choice:\n");
        scanf("%d", &ch);

        switch(ch)
        {
            Case 1: push();
                      break;

            Case 2: pop();
                      break;

            Case 3: display();
                      break;

            Case 4: / Exit(0);

            default:
                      printf("Invalid choice\n");
        }
    }
}
```

```
void push ()  
{  
    if (top == max - 1)  
    {  
        printf ("Stack overflow\n");  
    }  
    else  
    {  
        printf ("Enter the element to be added: ");  
        scanf ("%d", &item);  
        top = top + 1;  
        s[top] = item;  
    }  
}
```

```
void pop ()
```

```
{  
    if (top == -1)  
    {  
        printf ("Stack underflow\n");  
    }  
    else  
    {  
        printf ("The popped element: %d", s[top]);  
        top = top - 1;  
    }  
}
```

```
void display ()
```

```
{  
    if (top == -1)  
    {  
        printf ("Underflow!!\n");  
    }  
    else  
    {  
        printf ("Elements present in the stack:\n");  
        for (int i = top; i >= 0; i--)  
            printf ("%d\n", s[i]);  
    }  
}
```

Output :

Perform operations on the stack

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

Enter the choice : 1

Enter the element to be added onto the stack : 10

Perform operations on the stack

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

Enter the choice : 1

Enter the element to be added onto the stack : 20

Perform operations on the stack

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

Enter the choice : 1

Enter the element to be added onto the stack 30

Perform operations on the stack:

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

Enter the choice : 1

Overflow!!

Perform the operations on the stack:

- 1. Push
- 2. Pop
- 3. Display
- 4. End

Enter the choice : 2

popped element : 30.

Perform the operations on the stack:

- 1. Push
- 2. Pop
- 3. Display
- 4. End

Enter the choice : 2

popped element : 20

Perform the operations on the stack

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

Enter the choice : 2

popped element : 10.

Perform the operations on the stack

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

Enter the choice : 2

Underflow !!

Perform the operations on the stack

- 1. pop
- 2. push
- 3. display
- 4. Exit

Enter the choice : 3

Underflow !!

S. P. CB
S. P. CB
11/01/2011