```c
#include <stdio.h>
#define SIZE 20
int array[SIZE];        // declaration of array type variable.
int top1 = -1;
int top2 = SIZE;

//Function to push data into stack1
void push1 (int data)
{
// checking the overflow condition
  if (top1 < top2 - 1)
    {
      top1++;
      array[top1] = data;
    }
  else
    {
      printf ("Stack is full");
    }
}

// Function to push data into stack2
void push2 (int data)
{
// checking overflow condition
  if (top1 < top2 - 1)
    {
      top2--;
      array[top2] = data;
    }
  else
    {
      printf ("Stack is full..\n");
    }
}

//Function to pop data from the Stack1
void pop1 ()
{
```

```
40: // Checking the underflow condition
41:   if (top1 >= 0)
42:     {
43:       int popped_element = array[top1];
44:       top1--;
45:
46:       printf ("%d is being popped from Stack 1\n", popped_element)
47:     }
48:   else
49:     {
50:       printf ("Stack is Empty \n");
51:     }
52: }
53:
54: // Function to remove the element from the Stack2
55: void pop2 ()
56: {
57: // Checking underflow condition
58:   if (top2 < SIZE)
59:     {
60:       int popped_element = array[top2];
61:       top2--;
62:
63:       printf ("%d is being popped from Stack 1\n", popped_element)
64:     }
65:   else
66:     {
67:       printf ("Stack is Empty!\n");
68:     }
69: }
70:
71: //Functions to Print the values of Stack1
72: void display_stack1 ()
73: {
74:   int i;
75:   for (i = top1; i >= 0; --i)
76:     {
77:       printf ("%d ", array[i]);
78:     }
```

```
 79:   printf ("\n");
 80: }
 81:
 82: // Function to print the values of Stack2
 83: void display_stack2 ()
 84: {
 85:   int i;
 86:   for (i = top2; i < SIZE; ++i)
 87:     {
 88:       printf ("%d ", array[i]);
 89:     }
 90:   printf ("\n");
 91: }
 92:
 93: int main ()
 94: {
 95:   int ar[SIZE];
 96:   int i;
 97:   int num_of_ele;
 98:
 99:   printf ("We can push a total of 20 values\n");
100:
101:   //Number of elements pushed in stack 1 is 10
102:   //Number of elements pushed in stack 2 is 10
103:
104: // loop to insert the elements into Stack1
105:   for (i = 1; i <= 10; ++i)
106:     {
107:       push1 (i);
108:       printf ("Value Pushed in Stack 1 is %d\n", i);
109:     }
110: // loop to insert the elements into Stack2.
111:   for (i = 11; i <= 20; ++i)
112:     {
113:       push2 (i);
114:       printf ("Value Pushed in Stack 2 is %d\n", i);
115:     }
116:
117:   //Print Both Stacks
```

```
118:    display_stack1 ();
119:    display_stack2 ();
120:
121:    //Pushing on Stack Full
122:    printf ("Pushing Value in Stack 1 is %d\n", 11);
123:    push1 (11);
124:
125:    //Popping All Elements from Stack 1
126:    num_of_ele = top1 + 1;
127:    while (num_of_ele)
128:      {
129:        pop1 ();
130:        --num_of_ele;
131:      }
132:
133:    // Trying to Pop the element From the Empty Stack
134:    pop1 ();
135:
136:    return 0;
137: }
```