

//11. Write a C program to implement Stack operations such as PUSH, POP and PEEK

```
#include<stdio.h>
#include<stdlib.h>
#define MAXSIZE 5
struct stack
{
    int stk[MAXSIZE];
    int top;
};
typedef struct stack ST;
ST s;
/*Function to add an element to stack */
void push ()
{
    int num;
    if (s.top == (MAXSIZE - 1))
    {
        printf ("Stack is Full\n");
        return;
    }
    else
    {
        printf ("\nEnter element to be pushed : ");
        scanf ("%d", &num);
        s.top = s.top + 1;
        s.stk[s.top] = num;
    }
    return;
}
/*Function to delete an element from stack */
int pop ()
{
    int num;
    if (s.top == - 1)
    {
        printf ("Stack is Empty\n");
        return (s.top);
    }
    else
    {
        num = s.stk[s.top];
        printf ("poped element is = %d\n", s.stk[s.top]);
        s.top = s.top - 1;
    }
}
```

```

        return(num);
    }
/*Function to display the status of stack */
void display ()
{
    int i;
    if (s.top == -1)
    {
        printf ("Stack is empty\n");
        return;
    }
    else
    {
        printf ("\nStatus of elements in stack : \n");
        for (i = s.top; i >= 0; i--)
        {
            printf ("%d\n", s.stk[i]);
        }
    }
}
int main ()
{
    int ch;
    s.top = -1;

    printf ("\tSTACK OPERATIONS\n");
    printf ("-----\n");
    printf("    1. PUSH\n");
    printf("    2. POP\n");
    printf("    3. DISPLAY\n");
    printf("    4. EXIT\n");
    //printf("-----\n");
    while(1)
    {
        printf("\nChoose operation : ");
        scanf("%d", &ch);
        switch (ch)
        {
            case 1:
                push();
                break;
            case 2:
                pop();
                break;

```

```

        case 3:
            display();
        break;
        case 4:
            exit(0);
        default:
            printf("Invalid operation \n");
    }
}
return 0;
}

```

The screenshot shows a C++ IDE window titled "D:\data structures lab\push.pop on stack.c - [Executing] - Dev-C++ 5.11". The code is a C program implementing stack operations. The stack is represented as an array of integers, with a 'top' pointer indicating the current top of the stack. The 'push' function checks if the stack is full (top == MAXSIZE - 1) and prints an error if so. Otherwise, it prompts the user to enter an element to be pushed, reads it, and pushes it onto the stack. The 'pop' function checks if the stack is empty (top == -1) and prints an error if so, then returns the top element and increments the 'top' pointer.

```

1 //12.. Write a C program to implement Stack operations such as PUSH, POP and PEEK
2 #include<stdio.h>
3 #include<stdlib.h>
4 #define MAXSIZE 5
5 struct stack
6 {
7     int stk[MAXSIZE];
8     int top;
9 };
10 typedef struct stack ST;
11 ST s;
12 /*Function to add an element to stack */
13 void push ()
14 {
15     int num;
16     if (s.top == (MAXSIZE - 1))
17     {
18         printf ("Stack is Full\n");
19         return;
20     }
21     else
22     {
23         printf ("\nEnter element to be pushed : ");
24         scanf ("%d", &num);
25         s.top = s.top + 1;
26         s.stk[s.top] = num;
27     }
28     return;
29 }
30 /*Function to delete an element from stack */
31 int pop ()
32 {
33     int num;
34     if (s.top == - 1)
35     {
36         printf ("Stack is Empty\n");
37         return (s.top);
38     }
39 }

```

```
D:\data structures lab\push_pop on stack.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
[*] linked list operations.c push_pop on stack.c
34 if (s.top == -1)
35 {
36     printf ("Stack is Empty\n");
37     return (s.top);
38 }
39 else
40 {
41     num = s.stk[s.top];
42     printf ("popped element is = %d\n", s.stk[s.top]);
43     s.top = s.top - 1;
44 }
45 return(num);
46 }
47 /*Function to display the status of stack */
48 void display ()
49 {
50     int i;
51     if (s.top == -1)
52     {
53         printf ("Stack is empty\n");
54         return;
55     }
56     else
57     {
58         printf ("\nStatus of elements in stack : \n");
59         for (i = s.top; i >= 0; i--)
60         {
61             printf ("%d\n", s.stk[i]);
62         }
63     }
64 }
65 int main ()
66 {
67     int ch;
68     s.top = -1;
69     printf ("\tSTACK OPERATIONS\n");
70 }
Line: 21 Col: 11 Sel: 0 Lines: 99 Length: 2322 Insert Done parsing in 0 seconds
89°F Partly sunny
```

```
D:\data structures lab\push_pop on stack.c - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
[*] linked list operations.c push_pop on stack.c
64 }
65 int main ()
66 {
67     int ch;
68     s.top = -1;
69     printf ("\tSTACK OPERATIONS\n");
70     printf("-----\n");
71     printf("1. PUSH\n");
72     printf("2. POP\n");
73     printf("3. DISPLAY\n");
74     printf("4. EXIT\n");
75     //printf("-----\n");
76     while(1)
77     {
78         printf("\nChoose operation : ");
79         scanf("%d", &ch);
80         switch (ch)
81         {
82             case 1:
83                 push();
84                 break;
85             case 2:
86                 pop();
87                 break;
88             case 3:
89                 display();
90                 break;
91             case 4:
92                 exit(0);
93             default:
94                 printf("Invalid operation \n");
95         }
96     }
97     return 0;
98 }
99 }
Line: 21 Col: 11 Sel: 0 Lines: 99 Length: 2322 Insert Done parsing in 0 seconds
89°F Partly sunny
```

```
D:\data structures lab\push_pop on stack.exe
STACK OPERATIONS
1. PUSH
2. POP
3. DISPLAY
4. EXIT
Choose operation : 1
Enter element to be pushed : 10
Choose operation : 1
Enter element to be pushed : 43
Choose operation : 1
Enter element to be pushed : 34
Choose operation : 1
Enter element to be pushed : 39
Choose operation : 2
popped element is = 39
Choose operation : 3
Status of elements in stack :
34
43
10
Choose operation : 4
-----
Process exited after 57.16 seconds with return value 0
Press any key to continue . . .
```