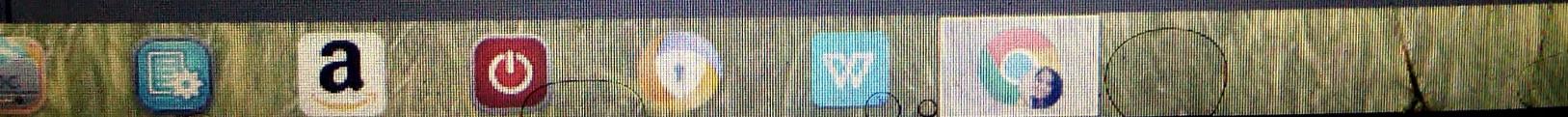


```
1 * #include <stdio.h>
2 #include <stdlib.h>
3 #define SIZE 5
4 int top=-1;
5 int stack[SIZE];
6 void push(int ele)
7 {
8     if(top==SIZE-1)
9     {
10         printf("The stack is overflow \n");
11     }
12     else
13     {
14         top++;
15         stack[top]=ele;
16     }
17 }
18 int pop()
19 {
20     if(top== -1)
21     {
22         return 0;
23     }
24     else
25     {
26         printf("Element removed is : %d\n",stack[top--]);
27         return 1;
28     }
29 }
```



```
--  
26         return 1;  
27     }  
28 }  
29  
30 void display()  
31 {  
32     if(top== -1)  
33         printf("The stack is empty\n");  
34     else  
35 {  
36         printf("The stack elements are\n");  
37         for(int i=0;i<=top;i++)  
38 {  
39             printf("%d\n",stack[i]);  
40         }  
41     }  
42 }  
43  
44 int main()  
45 {  
46     int c,d,p;  
47     while(c!=4)  
48 {  
49         printf("Enter command\tt1-push\tt2-pop\tt3-Display\tt4-Exit\n");  
50         scanf("%d",&c);  
51         switch(c)
```





Run code

```
46     int c,d,p;
47     while(c!=4)
48     {
49         printf("Enter command\t1-push\t2-pop\t3-Display\t4-Exit\n");
50         scanf("%d",&c);
51         switch(c)
52         {
53             case 1:printf("Enter an element to be inserted\n");
54                 scanf("%d",&d);
55                 push(d);
56                 break;
57             case 2:p=pop();
58                 if(p==0)
59                     printf("Stack is underflow\n");
60                 else
61                     printf("\nElement removed successfully\n");
62                 break;
63             case 3:display();
64                 break;
65             case 4:break;
66             default: printf("Invalid input\n");
67         }
68     }
69     return 0;
70 }
71
```



## X Output

Enter command 1-push 2-pop 3-Display 4-Exit  
">>>>1

Enter an element to be inserted  
>>>2

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>3

The stack elements are  
2

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>2

Element removed is : 2

Element removed successfully

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>2

Stack is underflow

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>1

Enter an element to be inserted  
>>>2

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>1

Enter an element to be inserted  
>>>2

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>1

Enter an element to be inserted  
>>>3

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>2

Element removed is : 3

Element removed successfully

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>1

Enter an element to be inserted

Element removed successfully  
Enter command 1-push 2-pop 3-Display 4-Exit  
>>>3

The stack elements are

2  
2

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>3

The stack elements are

2  
2

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>2

Element removed is : 2

Element removed successfully

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>3

The stack elements are

2

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>2

Element removed is : 2

Element removed successfully

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>1

Enter an element to be inserted

>>>4

Enter command 1-push 2-pop 3-Display 4-Exit  
>>>4

Process Finished.

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE 5
int top = -1;
int stack[SIZE];
void push(int ele)
{
    if (top == SIZE - 1)
    {
        printf ("The stack is full\n");
        ↳ overflow
    }
    else
    {
        top++;
        stack[top] = ele;
    }
}
int pop()
{
    if (top == -1)
    {
        return 0;
    }
    else
    {
        printf ("Element removed is: %.d\n", stack[top]);
    }
}
```

DATE: / /

```
return 1;  
}  
}  
  
void display()  
{  
    if (top == -1)  
        printf ("The stack is empty\n");  
    else  
    {  
        printf ("The elements are\n");  
        for (int i=0; i<=top; i++)  
        {  
            printf ("%d\n", stack[i]);  
        }  
    }  
}  
  
int main()  
{  
    int c, d, p;  
    while (c!=4)  
    {  
        printf ("Enter command 1- push 2-pop  
                3-Display 4-Exit\n");  
        scanf ("%d", &c);  
        if (c==1)  
        {  
            printf ("Enter element\n");  
            scanf ("%d", &d);  
            push(d);  
        }  
        else if (c==2)  
        {  
            pop();  
        }  
        else if (c==3)  
        {  
            display();  
        }  
        else if (c==4)  
        {  
            break;  
        }  
    }  
}
```

```
case 1: printf ("Enter an element\n");
scanf ("%d", &d);
push(d);
break;

case 2: p = pop();
if (p == 0)           underflow
printf ("Stack is empty\n");
else
printf ("An element removed \n");
break;

case 3: display();
break;

case 4: break;

default: printf ("Invalid input\n");

}

}

return 0;
}
```