

```

9  #include <stdio.h>
10 #include <stdlib.h>
11 #define STACK_SIZE 3
12 int top=-1,s[10],item;
13 void push()
14 {
15     if(top==STACK_SIZE-1)
16     {
17         printf("STACK OVERFLOW \n");
18         return;
19     }
20     else
21     {
22         top=top+1;
23         s[top]=item;
24     }
25 }
26
27 int pop()
28 {
29     if(top== -1)
30     {
31         printf("STACK UNDERFLOW \n");
32         return -1;
33     }
34     else
35         return s[top--];
36 }
37 void display()

```

```

38 {
39     int i;
40     if(top==1)
41     {
42         printf("STACK is empty \n");
43         return;
44     }
45     else
46         printf("Contents of the stack are\n");
47     for(i=top;i>0;i--)
48         printf("%d \n",s[i]);
49 }
50 int main()
51 {
52     int item_deleted;
53     int choice;
54     for(;;)
55     {
56         printf("Enter choice: \n1.Push \n2.Pop \n3.Display \n4.Exit \n");
57         scanf("%d",&choice);
58         switch(choice)
59         {
60             case 1:
61                 printf("Enter the item to be pushed:\n ");
62                 scanf("%d",&item);
63                 push();
64                 break;
65             case 2:
66                 item_deleted=pop();

```

```

67         if(item_deleted--1)
68             printf("STACK is empty \n");
69         else
70             printf("The item deleted is: %d \n",item_deleted);
71         break;
72     case 3:
73         display();
74         break;
75     default: exit(0);
76     }
77 }
78 return 0;
79 }
80

```

input

```

Enter choice:
1.Push
2.Pop
3.Display
4.Exit
1
Enter the item to be pushed:
1
Enter choice:
1.Push
2.Pop
3.Display
4.Exit
1
Enter the item to be pushed:

```

Enter the item to be pushed:

2

Enter choice:

1.Push

2.Pop

3.Display

4.Exit

1

Enter the item to be pushed:

3

Enter choice:

1.Push

2.Pop

3.Display

4.Exit

1

Enter the item to be pushed:

4

STACK OVERFLOW

Enter choice:

1.Push

2.Pop

3.Display

4.Exit

3

Contents of the stack are

3

2

1

Enter choice:

1.Push

```
2.Pop
3.Display
4.Exit
2
The item deleted is: 3
Enter choice:
1.Push
2.Pop
3.Display
4.Exit
2
The item deleted is: 2
Enter choice:
1.Push
2.Pop
3.Display
4.Exit
2
The item deleted is: 1
Enter choice:
1.Push
2.Pop
3.Display
4.Exit
2
STACK UNDERFLOW
STACK is empty
Enter choice:
1.Push
2.Pop
3.Display
```

WAP to simulate the working of stack using an array with following a push pop & display.

```
#include <stdio.h>
```

```
#define STACK_SIZE 10
```

```
int top = -1, S[10], item;
```

```
void push(int item, int S, int top)
```

```
{  
    if (top == STACK_SIZE - 1)
```

```
    printf("Stack overflow \n");  
    return 0;
```

```
    top = top + 1;
```

```
    S[top] = item;
```

```
}  
int pop()
```

```
{  
    if (top == -1)
```

```
    printf("Stack underflow \n");  
    return -1;
```

```
    item = S[top];
```

```
    return S[top--];
```

```
}
```

```
void display()
```

```
{
```

```

int i;
if (top == -1)
{
    printf("Stack is empty\n");
    return;
}
else
{
    printf("Contents in the stack\n");
    for (i = top; i >= 0; i--)
        printf("%d", S[i]);
}

```

```

void main()

```

```

{
    int item-deleted;

```

```

    int choice;

```

```

    for(;;)
    {

```

```

        printf("Enter choice\n 1. Push\n 2. Pop\n 3. display\n 4. exit\n");

```

```

        scanf("%d", &choice);

```

```

        switch(choice)
        {

```

```

            case 1:

```

```

                printf("Enter the item to be pushed\n");

```

```

                scanf("%d", &item);

```

```

                push();

```

```

                break;

```

```

            case 2: item-deleted = pop();

```

```

                if (item-deleted == -1)

```

```

                    printf("Stack is empty\n");

```

```

                }
            }
}

```



8

else

printf("The item deleted is : %d\n",  
item-deleted);

break;

case 3 : display();

break;

default : exit 0;

}

}

getch();

}