

```

8
9 #include <stdio.h>
10 #include <stdlib.h>
11 #define STACK_SIZE 3
12 int s[3], item;
13 int top = -1;
14 void push()
15 {
16     if(top == STACK_SIZE - 1)
17     {
18         printf("stack overflow\n");
19     }
20     top = top + 1;
21     s[top] = item;
22 }
23 int pop()
24 {
25     if(top == -1){
26         printf("stack underflow\n");
27     }
28     return s[top = top - 1];
29 }
30 void display()
31 {
32     int i;
33     if(top == -1){
34         printf("Empty stack\n");
35     }
36     printf("contents of the stack are empty\n");
37     for(i=0; i <= top; i++){
38         printf("%d\n", s[i]);
39     }
40 }
41 void main()
42 {
43     int item_deleted;
44     int choice=1;
45     while(choice != 0)
46     {
47         printf("1.push\n2.pop\n3.display\n4.exit\n");
48         printf("Enter choice\n");

```

input

```

31 - {
32     int i;
33     if(top == -1){
34         printf("Empty stack\n");
35     }
36     printf("contents of the stack are empty\n");
37     for(i=0; i <= top; i++){
38         printf("%d\n",s[i]);
39     }
40 }
41 void main()
42 {
43     int item_deleted;
44     int choice=1;
45     while(choice != 0)
46     {
47         printf("1.push\n2.pop\n3.display\n4.exit\n");
48         printf("Enter choice\n");
49         scanf("%d",&choice);
50         switch (choice){
51             case 1: printf("Enter number to be inserted\n");
52                     scanf("%d",&item);
53                     push();
54                     break;
55             case 2: item_deleted = pop();
56                     if(item_deleted != 0){
57                         printf("item deleted is %d\n",item_deleted);
58                         break;
59                     }
60             else{
61                 printf("Stack underflow\n");
62             }
63             case 3: display();
64                     break;
65             case 4: choice = 0;
66                     break;
67             default: printf("Invalid input\n");
68         }
69     }
70 }
71

```



```
1.push
2.pop
3.display
4.exit
Enter choice
1
Enter number to be inserted
2
1.push
2.pop
3.display
4.exit
Enter choice
1
Enter number to be inserted
3
1.push
2.pop
3.display
4.exit
Enter choice
1
Enter number to be inserted
4
1.push
2.pop
3.display
< 4.exit
Enter choice
1
Enter number to be inserted
6
stack overflow
1.push
2.pop
3.display
4.exit
Enter choice
3
contents of the stack are empty
2
3
4
6
1.push
2.pop
3.display
4.exit
Enter choice
2
item deleted is 4
1.push
2.pop
```

I

```
6
stack overflow
1.push
2.pop
3.display
4.exit
Enter choice
3
contents of the stack are empty
2
3
4
6
1.push
2.pop
3.display
4.exit
Enter choice
2
item deleted is 4
1.push
2.pop
3.display
4.exit
Enter choice
2
item deleted is 3
< 1.push
2.pop
3.display
4.exit
Enter choice
2
item deleted is 2
1.push
2.pop
3.display
4.exit
Enter choice
2
stack underflow
Empty stack
contents of the stack are empty
1.push
2.pop
3.display
4.exit
Enter choice
4
```

```

8
9 #include <stdio.h>
10 #include <stdlib.h>
11 #define STACK_SIZE 3
12 int top = -1;
13 void push(int item, int s[3], int *top)
14 {
15     if (*top == STACK_SIZE-1)
16     {
17         printf("stack overflow\n");
18         return ;
19     }
20     *top = *top + 1;
21     s[*top] = item;
22 }
23 int pop(int s[], int *top)
24 {
25     int item_deleted;
26     if(*top == -1)
27     {
28         printf("stack underflow\n");
29     }
30     item_deleted = s[*top];
31     *top = *top - 1;
32     return item_deleted;
33 }
34 int display(int top, int s[3])
35 {
36     int i;
37     if(top == -1)
38     {
39         printf("it is an empty stack\n");
40         return 0;
41     }
42     printf("The stack is\n");
43     for(i=0; i <= top; i++)
44     {
45         printf("%d\n", s[i]);
46     }
47     return 0;

```



```

38- {
39-     printf("it is an empty stack\n");
40-     return 0;
41- }
42- printf("The stack is\n");
43- for(i=0; i <= top; i++)
44- {
45-     printf("%d\n",s[i]);
46- }
47- return 0;
48- }
49- void main()
50- {
51-     int item,s[3];
52-     int item_deleted;
53-     int choice=1;
54-     while(choice != 0){
55-     {
56-         printf("1.push\n2.pop\n3.display\n4.exit\n");
57-         printf("Enter number\n");
58-         scanf("%d",&choice);
59-     }
60-     switch (choice){
61-         case 1: printf("Enter number to be inserted\n");
62-             scanf("%d",&item);
63-             push(item,s,&top);
64-             break;
65-         case 2: item_deleted = pop(s,&top);
66-             if(item_deleted != 0){
67-                 printf("item deleted is %d\n",item_deleted);
68-                 break;
69-             }
70-         case 3: display(top,s);
71-             break;
72-         case 4: choice = 0;
73-             break;
74-         default:printf("invalid input\n");
75-     }
76- }
77- }

```

Input

```
1.push
2.pop
3.display
4.exit
Enter number
1
Enter number to be inserted
2
1.push
2.pop
3.display
4.exit
Enter number
1
Enter number to be inserted
3
1.push
2.pop
3.display
4.exit
Enter number
1
Enter number to be inserted
4
1.push
2.pop
< 3.display
4.exit
Enter number
1
Enter number to be inserted
6
stack overflow
1.push
2.pop
3.display
4.exit
Enter number
3
The stack is
2
3
4
1.push
2.pop
3.display
4.exit
Enter number
2
item deleted is 4
1.push
2.pop
```

Enter number

3

The stack is

2

3

4

1.push

2.pop

3.display

4.exit

Enter number

2

item deleted is 4

1.push

2.pop

3.display

4.exit

Enter number

2

item deleted is 3

1.push

2.pop

3.display

4.exit

Enter number

2

< item deleted is 2

I

1.push

2.pop

3.display

4.exit

Enter number

2

stack underflow

item deleted is 2

1.push

2.pop

3.display

4.exit

Enter number

2

item deleted is 2

1.push

2.pop

3.display

4.exit

Enter number

4