

```

9  #include <stdio.h>
10 #include<stdlib.h>
11 #define STACK_SIZE 3
12 int stack[STACK_SIZE],TOP;
13 void display(int stack[])
14 {
15     int i=0;
16     if(TOP==-1)
17     {
18         printf("Stack underflow.\n");
19         return;
20     }
21     printf("%d",stack[TOP]);
22     for(i=TOP-1;i >=0;i--)
23     {
24         printf("\n%d",stack[i]);
25     }
26     printf("\n\n");
27 }
28 void PUSH(int stack[],int item)
29 {
30     if(TOP==STACK_SIZE-1)
31     {
32         printf("\nStack overflow\n");
33         return;
34     }
35     TOP++;
36     stack[TOP]=item;
37 }
38 void POP (int stack[])
39 {

```



```

}
void POP (int stack[])
{
    int deletedItem;
    if(TOP== -1)
    {
        printf("Stack underflow\n");
        return;
    }
    deletedItem=stack[TOP];
    TOP--;
    printf("%d deleted successfully\n",deletedItem);
    return;
}

void main()
{
    int item=0;
    int choice=0;
    TOP=-1;

    while(1)
    {
        printf("Enter Choice (1: display, 2: insert (PUSH), 3: remove (POP), 4: Exit...:");
        scanf("%d",&choice);

        switch(choice)
        {
            case 1:
                display(stack);
                break;
            case 2:
                push(stack);
                break;
            case 3:
                POP(stack);
                break;
            case 4:
                exit(0);
                break;
        }
    }
}

```

```

8 TOP=-1;
9
10 while(1)
11 {
12     printf("Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...");
13     scanf("%d",&choice);
14
15     switch(choice)
16     {
17         case 1:
18             display(stack);
19             break;
20         case 2:
21             printf("Enter Item to be insert :");
22             scanf("%d",&item);
23             PUSH(stack,item);
24             break;
25         case 3:
26             POP(stack);
27             break;
28         case 4:
29             exit(0);
30         default:
31             printf("\nInvalid choice.");
32             break;
33     }
34 }
35
36

```

```
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:2
Enter Item to be insert :10
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:2
Enter Item to be insert :20
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:2
Enter Item to be insert :30
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:2
Enter Item to be insert :40
```

Stack overflow

```
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:1
30
20
10
```

```
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:3
30 deleted successfully
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:3
20 deleted successfully
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:3
10 deleted successfully
```

```
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:3
Stack underflow
```

```
Enter Choice (1: display, 2: insert (PUSH), 3: remove(POP)), 4: Exit...:4
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```


18. Write a program to simulate the working of stack using an array with the following a: push, pop, display. The program should print appropriate msgs for stack overflow, stack underflow.

```
#include <stdio.h>
#define STACK_SIZE MAX 2
int stack [STACK_SIZE MAX], TOP;
void display (int stack[])
{
    int i = 0;
    if (TOP == -1) {
        printf("stack underflow");
        return;
    }
    printf(" %d", stack[TOP]);
    for (i = TOP - 1; i >= 0; i--) {
        printf(" %d", stack[i]);
    }
    printf("\n");
}
```

```
void push (int stack[], int item)
```

```
{  
    if (Top == STACK_SIZE - 1)  
    {  
        printf ("Stack overflow");  
        return;  
    }  
    Top++;  
    stack[Top] = item;
```

```
}
```

```
void pop (int stack[]) {
```

```
    int deleted_item;
```

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

```
        printf ("Stack underflow\n");
```

```
        return;
```

```
}
```

```
    deleted_item = stack[Top];
```

```
    Top--;
```

```
    printf ("%d deleted successfully\n", deleted_item);
```

```
    return;
```

```
}
```

```
void main() {
```

```
    int item = 0;
```

```
    int choice = 0;
```

```
    int Top = -1
```

~~write~~ for(;;) {

printf("Enter choice (1: display, 2: insert (push)

3: remove (pop), 4: Exit: ");

scanf("%d", &choice);

switch (choice) {

case 1: display (stack);

break;

case 2: printf("Enter item to be inserted: ");

scanf("%d", &item);

push (stack, item);

break;

case 3: pop (stack);

break;

case 4: exit (0);

default: printf("Invalid input");

break;

}

}

output:

Enter choice (1: display, 2: insert, 3: pop, 4: exit): 2

Enter item to be inserted: 10

Enter choice (1: display, 2: insert, 3: pop, 4: exit): 2

Enter item to be inserted: 20.

Enter choice (1: display, 2: insert, 3: pop, 4: exit): 2

Stack overflow

Enter choice (1: display, 2: insert, 3: pop, 4: exit): 1

20

10

Enter choice (1: display, 2: insert, 3: pop, 4: exit): 3

20 deleted successfully.

Enter choice (1: display, 2: insert, 3: pop, 4: exit): 3

10 deleted successfully.

Enter choice (1: display, 2: insert, 3: pop, 4: exit): 3

Stack underflow

Enter choice (1: display, 2: insert, 3: pop, 4: exit): 4

... Program finished with exit code 0.