

```

#include <stdio.h>
#include <conio.h>
#define stacksize 5
int top = -1;
int s[5];
int item;
void push()
{
    if (top == stacksize - 1)
    {
        printf("stack overflow\n");
        return;
    }
}

```

```

{
    top = top + 1;
    s[top] = item;
}

```

```

}
int pop()
{
    if (top == -1) return -1;
    return s[top--];
}

```

```

}
void display()
{
    int i;
    if (top == -1)
    {

```

```
printf ("stack is empty \n");  
return ;
```

```
}
```

```
printf ("contents of the stack \n");
```

```
for (i=0; i<=top; i++)
```

```
{
```

```
printf ("%d", s[i]);
```

```
}
```

```
}
```

```
void main ()
```

```
{
```

```
int item_delete;
```

```
int choice;
```

```
while (1)
```

```
{
```

```
printf ("\n 1. push \n 2. pop \n 3. display ");
```

```
printf ("enter the choice \n");
```

```
scanf ("%d", &choice);
```

```
switch (choice)
```

```
{
```

```
case 1: printf ("enter item to be inserted \n");
```

```
scanf ("%d", &item);
```

```
break;
```

```
Case 2: item_delete = pop();  
if (item_delete == -1)  
printf("stack is empty\n");  
else  
printf("item deleted is %d\n", item_delete);  
break;
```

```
Case 3: display();
```

```
break;
```

```
default: printf("invalid\n");  
break;
```

```
y
```

```
y
```

```
getch();
```

```
y
```

```

#include<stdio.h>
#include<conio.h>
#define stacksize 5
int top=-1;
int s[10];
int item;
void push()
{
    if(top==stacksize-1)
    {
        printf("stack overflow \n");
        return ;
    }
    top=top+1;
    s[top]=item;
}
int pop()
{
    if(top== -1) return -1;
    return s[top--];
}
void display()
{
    int i;
    if(top== -1)
    {
        printf("stack is empty \n");
        return ;
    }
}

```

```

    printf("stack is empty \n");
    return ;
}
printf("contents of the stack \n");
for(i=0;i<=top;i++)
{
    printf("%d",s[i]);
}
}

void main()
{
    int item_delete;
    int choice;
    while(1)
    {
        printf("\n 1.push \n 2.pop \n 3.display  " );
        printf("enter the choice \n ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: printf("enter item to be inserted \n");
                    scanf("%d",&item);
                    push(item);
                    break;
            case 2: item_delete=pop();
                    if(item_delete==-1)
                        printf("stack is empty \n");
                    else
                        printf("item deleted is %d \n",item_delete);
        }
    }
}

```



```

main.c x
39 int item_delete;
40 int choice;
41 while(1)
42 {
43     printf("\n 1.push \n 2.pop \n 3.display  " );
44     printf("enter the choice \n ");
45     scanf("%d",&choice);
46     switch(choice)
47     {
48         case 1: printf("enter item to be inserted \n");
49                 scanf("%d",&item);
50                 push(item);
51                 break;
52         case 2: item_delete=pop();
53                 if(item_delete==1)
54                     printf("stack is empty \n");
55                 else
56                     printf("item deleted is %d \n",item_delete);
57                 break;
58         case 3: display();
59                 break;
60
61         default: printf("invalid \n");
62                 break;
63     }
64     getch();
65 }
66
67

```

```
1.push
2.pop
3.display enter the choice
1
enter item to be inserted
0

1.push
2.pop
3.display enter the choice
1
enter item to be inserted
0

1.push
2.pop
3.display enter the choice
enter item to be inserted
8

1.push
2.pop
3.display enter the choice
34
invalid

1.push
2.pop
3.display enter the choice
2
item deleted is 78

1.push
2.pop
3.display enter the choice
3
contents of the stack
030

1.push
2.pop
3.display enter the choice
```