

Program - 1

```
#include <stdio.h>
#include <stdlib.h>
#define Max 05
int stack[Max], Top;
void display(int stack[])
{
    int i = 0;
    if (Top == -1)
    {
        printf("Stack is Empty \n");
        return;
    }
    printf("%d <-- TOP", stack[Top]);
    for (i = Top - 1; i >= 0; i--)
    {
        printf("\n %d", stack[i]);
    }
    printf("\n \n");
}

void Push(int stack[], int item)
{
    if (Top == Max - 1)
    {
        printf("\n Stack is full Can't add item \n");
    }
}
```

```
return;
```

```
}
```

```
TOP++;
```

```
stack[TOP] = item;
```

```
}
```

```
void POP(int stack[])
```

```
{
```

```
    int deleted item;
```

```
    if (TOP == -1)
```

```
    {
```

```
        Print ("Stack is empty \n");
```

```
        return;
```

```
    }
```

```
    deleted item = stack[TOP];
```

```
    TOP --;
```

```
    printf ("%d deleted Successfully \n", deleted item);
```

```
    return;
```

```
}
```

```
void main()
```

```
{
```

```
    int item = 0;
```

```
    int choice = 0;
```

```
    TOP = -1;
```

```
    while(1)
```



```
{  
    printf ("Enter choice (1: display, 2: insert (Pos#),  
        3: remove (POP), 4: exit);  
    scanf ("%d", &choice);
```

```
    switch(choice)
```

```
{
```

```
    case 1:
```

```
        display (stack);
```

```
        break;
```

```
    case 2:
```

```
        printf ("Enter item to be insert:");
```

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

```
        push (stack, item);
```

```
        break;
```

```
    case 3:
```

```
        pop (stack);
```

```
        break;
```

```
    case 4:
```

```
        exit (0);
```

```
    default:
```

```
        printf ("\n invalid choice");
```

```
        break;
```

```
} } getch (); }
```