

```
3
4 typedef struct {
5     int size;
6     int top;
7     int *s;
8     int *minstack;
9 }
10 } MinStack;
11
12
13 MinStack* minStackCreate() {
14     MinStack *st=(MinStack*) malloc(sizeof(MinStack));
15     if(st==NULL)
16     {
17         printf("memory allocation failed");
18         exit(0);
19     }
20     st->size=5;
21     st->top=-1;
22     st->s=(int*) malloc (st->size*sizeof(int));
23     st->minstack = (int*) malloc (st->size * sizeof(int));
24     if(st->s==NULL)
25     {
26         printf("memory allocation failed");
27         free(st->s);
28         free(st->minstack);
29         exit(0);
30     }
31     return st;
32 }
33
34
35 void minStackPush(MinStack* obj, int val) {
36     if(obj->top==obj->size-1)
37     {
38         printf("stack is overflow");
39     }
40     else{
41         obj->top++;
42         obj->s[obj->top]=val;
43     }
```

```

}
else{
    obj->top++;
    obj->s[obj->top]=val;
    if (obj->top == 0 || val < obj->minstack[obj->top - 1]) {
        obj->minstack[obj->top] = val;
    } else {
        obj->minstack[obj->top] = obj->minstack[obj->top - 1];
    }
}
}

```

```

}

void minStackPop(MinStack* obj) {
    int value;
    if(obj->top== -1)
    {
        printf("underflow");
    }
    else
    {
        value=obj->s[obj->top];
        obj->top--;
        printf("%d is popped\n",value);
    }
}

```

```

int minStackTop(MinStack* obj) {
    int value=-1;
    if(obj->top== -1)
    {
        printf("underflow\n");
        exit(0);
    }
    else
    {

```

```

78     else
79     {
80         value=obj->s[obj->top];
81         return value;
82     }
83 }
84
85 }
86
87 int minStackGetMin(MinStack* obj) {
88     if(obj->top==1)
89     {
90         printf("underflow\n");
91         exit(0);
92     }
93     else
94     {
95         return obj->minstack[obj->top];
96     }
97 }
98
99 }
100
101 void minStackFree(MinStack* obj) {
102     free(obj->s);
103     free(obj->minstack);
104     free(obj);
105 }
106
107
108
109 /**
110  * Your MinStack struct will be instantiated and called as such:
111  * MinStack* obj = minStackCreate();
112  * minStackPush(obj, val);
113
114  * minStackPop(obj);
115
116  * int param_3 = minStackTop(obj);
117
118  * int param_4 = minStackGetMin(obj);

```

Input

```
["MinStack","push","push","push","getMin","pop","top","getMin"]
```

```
[[],[-2],[0],[-3],[],[],[],[]]
```

Stdout

```
-3 is popped
```

Output

```
[null,null,null,null,-3,null,0,-2]
```

Expected

```
[null,null,null,null,-3,null,0,-2]
```