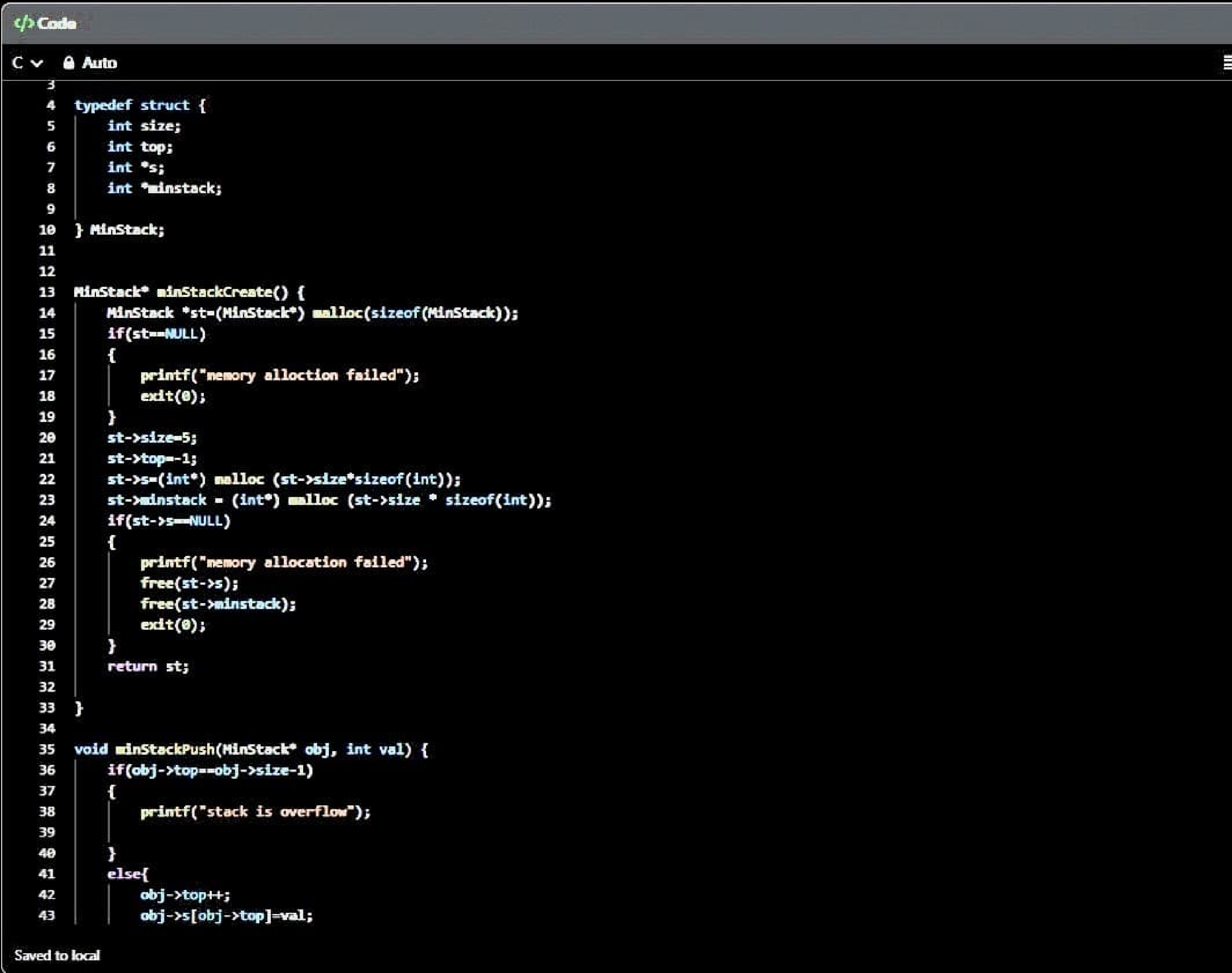
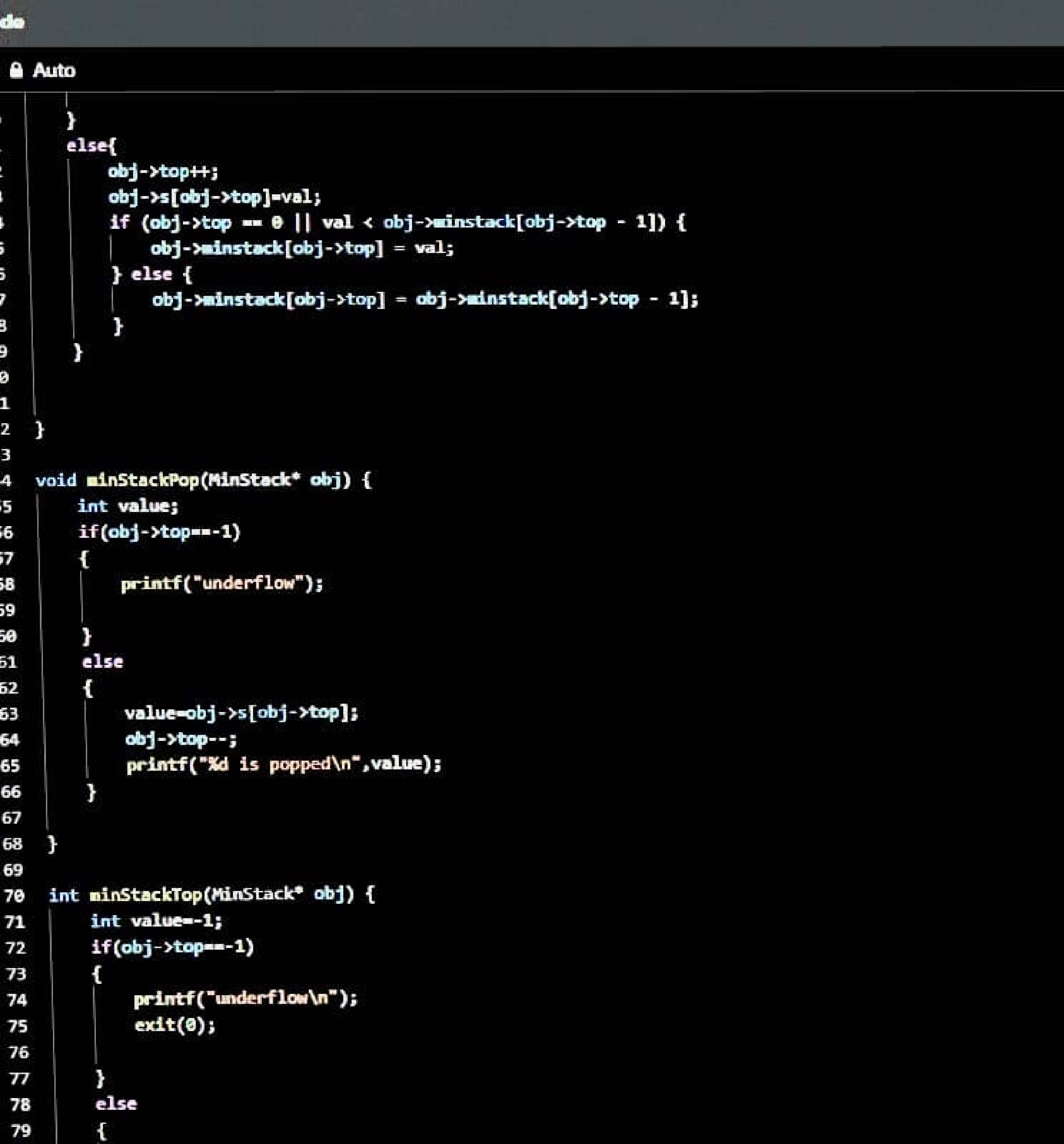
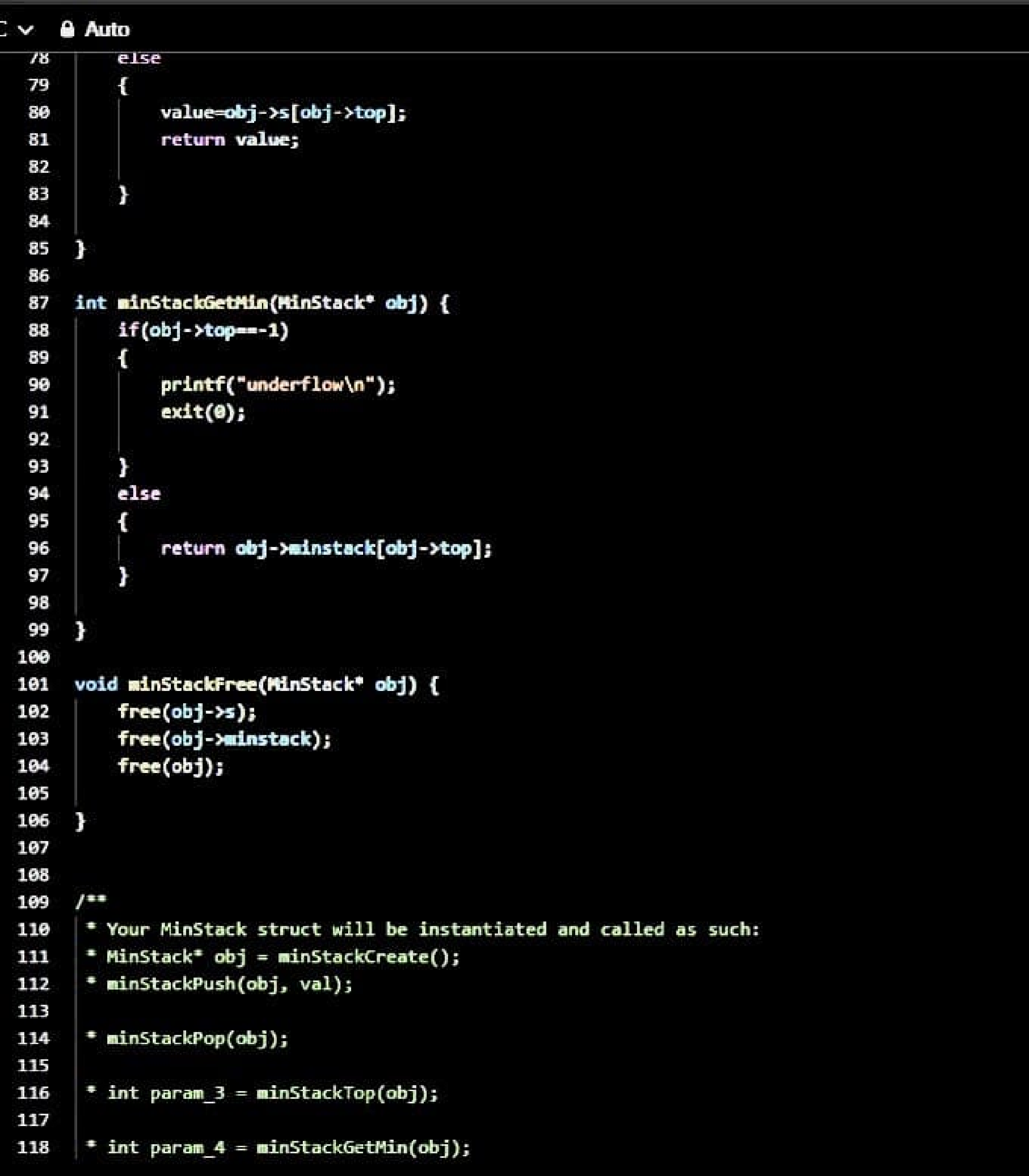


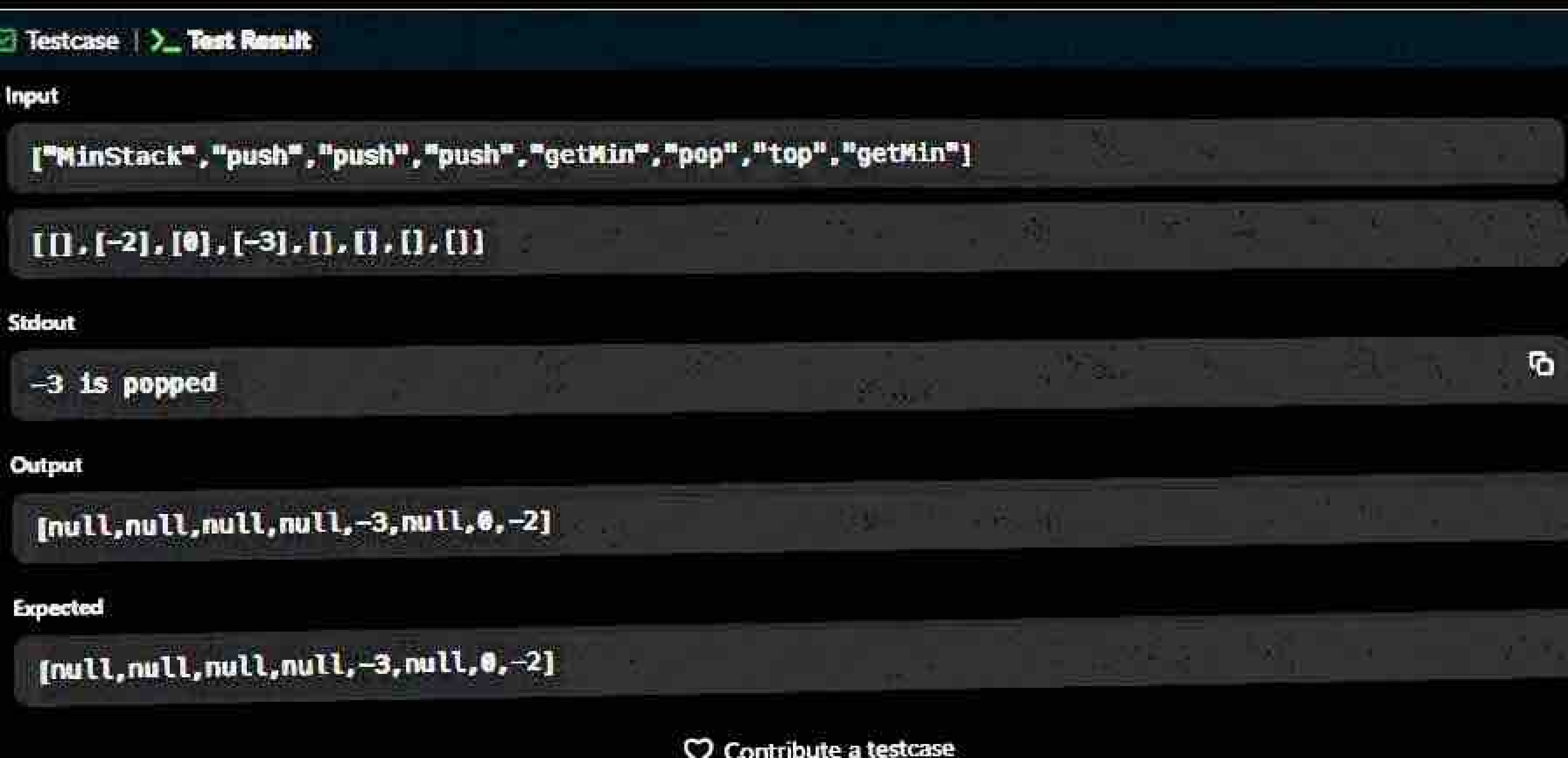
```c

typedef struct {
    int size;
    int top;
    int *s;
    int *minstack;

} MinStack;


MinStack* minStackCreate() {
    MinStack *st=(MinStack*) malloc(sizeof(MinStack));
    if(st==NULL)
    {
        printf("memory alloction failed");
        exit(0);
    }
    st->size=5;
    st->top=-1;
    st->s=(int*) malloc (st->size*sizeof(int));
    st->minstack = (int*) malloc (st->size * sizeof(int));
    if(st->s==NULL)
    {
        printf("memory allocation failed");
        free(st->s);
        free(st->minstack);
        exit(0);
    }
    return st;

}

void minStackPush(MinStack* obj, int val) {
    if(obj->top==obj->size-1)
    {
        printf("stack is overflow");

    }
    else{
        obj->top++;
        obj->s[obj->top]=val;
```



```
    }
    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
    {
```



```c
    else
    {
        value=obj->s[obj->top];
        return value;

    }

}

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

    }
    else
    {
        return obj->minstack[obj->top];
    }

}

void minStackFree(MinStack* obj) {
    free(obj->s);
    free(obj->minstack);
    free(obj);

}


/**
 * Your MinStack struct will be instantiated and called as such:
 * MinStack* obj = minStackCreate();
 * minStackPush(obj, val);

 * minStackPop(obj);

 * int param_3 = minStackTop(obj);

 * 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]
```