

[Description](#)
[Editorial](#)
[Solutions \(20.3K\)](#)
[Submissions](#)

## 20. Valid Parentheses

Hint

Easy
23K
1.6K

Companies

Given a string `s` containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

Testcase Result

### Compile Error

```

Line 64: Char 5: error: redefinition of 'main' [solution.c]
67 | int main(int argc, char *argv[]) {
    |      ^~~~~

```

Console



Run

Submit

i C | Auto









```

1  #include <stdio.h>
2  #include <stdbool.h>
3  #define N 100
4
5  char st[N];
6  int top = -1;
7
8  void push(char a);
9  char pop();
10 bool right(char s[]);
11
12 void push(char a) {
13     if (top == -1) {
14         printf("overflow");
15         return;
16     }
17     st[++top]=a;
18 }
19
20 char pop() {
21     if (top == -1) {
22         printf("empty");
23         return '\0';
24     }
25     char a = st[top];
26     --top;
27     return a;
28 }

```

Saved to local

Ln 17, Col 17

## 20. Valid Parentheses

Easy 23K 1.6K

Companies

Given a string `s` containing just the characters `'('`, `')'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

Testcase Result

### Compile Error

Line 64: Char 5: error: redefinition of 'main' [solution.c]  
67 | int main(int argc, char \*argv[]) {  
| ~~~~~

Console



Run

Submit

Dynamic Layout

Premium



i C Auto

{} ↺ ⌘ ⚙ ↻

```
22 char pop() {
23     if (top == -1) {
24         printf("empty");
25         return '\0';
26     }
27     char a = st[top];
28     top = top - 1;
29     return a;
30 }
31
32 bool right(char s[]) {
33     int i = 0;
34     while (s[i] != '\0') {
35         if (s[i] == '(' || s[i] == '{' || s[i] == '[') {
36             push(s[i]);
37         } else if (s[i] == ')' || s[i] == '}' || s[i] == ']') {
38             if (top == -1) {
39                 return false;
40             }
41             char temp = pop();
42             if ((temp == '(' && s[i] != ')') || (temp == '{' && s[i] != '}') ||
(temp == '[' && s[i] != ']')) {
43                 return false;
44             }
45         }
46         i++;
47     }
48
49     return top == -1;
```

Saved to local

Ln 61, Col 1

## 20. Valid Parentheses

Easy 23K 1.6K

Companies

Given a string `s` containing just the characters `'('`, `)'`, `'{'`, `'}'`, `'['` and `']'`, determine if the input string is valid.

Testcase Result

### Compile Error

```
Line 64: Char 5: error: redefinition of 'main' [solution.c]
67 | int main(int argc, char *argv[]) {
    |
```

Console

Run Submit

```

35     if (s[i] == '(' || s[i] == '{' || s[i] == '[') {
36         push(s[i]);
37     } else if (s[i] == ')' || s[i] == '}' || s[i] == ']') {
38         if (top == -1) {
39             return false;
40         }
41         char temp = pop();
42         if ((temp == '(' && s[i] != ')') || (temp == '{' && s[i] != '}') ||
43             (temp == '[' && s[i] != ']')) {
44             return false;
45         }
46         i++;
47     }
48     return top == -1;
49 }
50
51 int main() {
52     char s[] = "(";
53     if (right(s))
54         printf("Valid\n");
55     else
56         printf("Invalid\n");
57     return 0;
58 }
59
60
61
```

Saved to local

Ln 48, Col 5