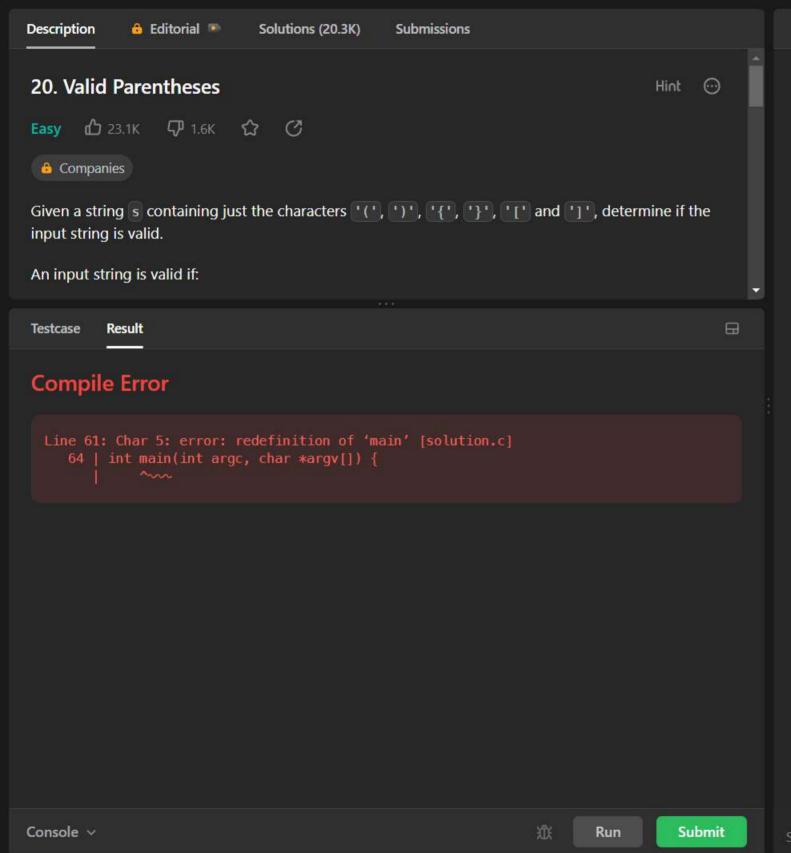


```
素 ② 第 ♂ {} □ ■
1 #include <stdio.h>
    #include <stdbool.h>
    #define N 100
    char st[N];
   int top = -1;
    void push(char a);
    char pop();
    bool right(char s[]);
    void push(char a) {
12
        if (top == -1) {
            printf("overflow");
            return;
        st[++top]=a;
17
    char pop() {
        if (top == -1) {
            printf("empty");
            return '\0';
        char a = st[top];
        --top;
        return a;
    bool right(char s[]) {
        int i = 0;
        while (s[i] != '\0') {
            if (s[i] == '(' || s[i] == '{' || s[i] == '[') {
               push(s[i]);
            } else if (s[i] == ')' || s[i] == '}' || s[i] == ']') {
               if (top == -1) {
                   return false;
```

Saved to local Ln 17, Col 17



```
素 ② 第 ♂ {} □ ■
        Auto
i C V
        --top;
        return a;
    bool right(char s[]) {
        int i = 0;
        while (s[i] != '\0') {
            if (s[i] == '(' || s[i] == '{' || s[i] == '[') {
                push(s[i]);
            } else if (s[i] == ')' || s[i] == '}' || s[i] == ']') {
                if (top == -1) {
                   return false;
                char temp = pop();
                if ((temp == '(' && s[i] != ')') || (temp == '{' && s[i] != '}') ||
    (temp == '[' && s[i] != ']')) {
                   return false;
            i++;
        return top==-1;
    int main() {
        char s[] = "({[]})";
        if (right(s))
            printf("Valid\n");
        else
            printf("Invalid\n");
        return 0;
58
```

Saved to local Ln 58, Col 1