

Valid Parenthesis

'(, ')', '{', '}', '[', ']'

Input 1

() [] { } = = = = = =

Output

true

Input 2

(]

Output

false

Input 3

{ [()] }).

Output

true

Input 4

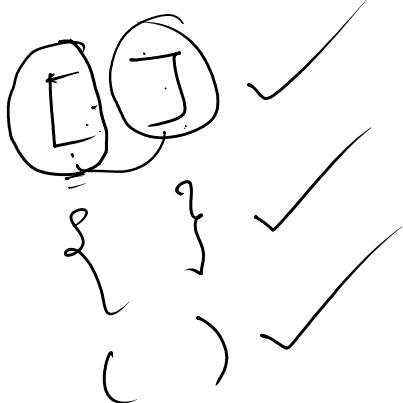
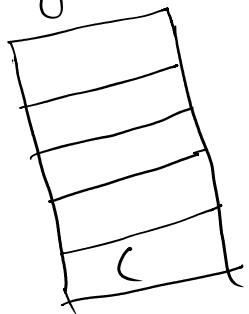
({ [{ }])

Output
false

Solution:-

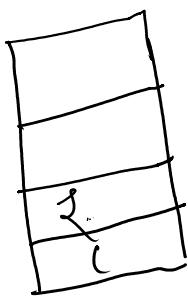
({ [] })

Opening characters \rightarrow '(', '{', '['



Ex2.

({ [] }) } \rightarrow not valid

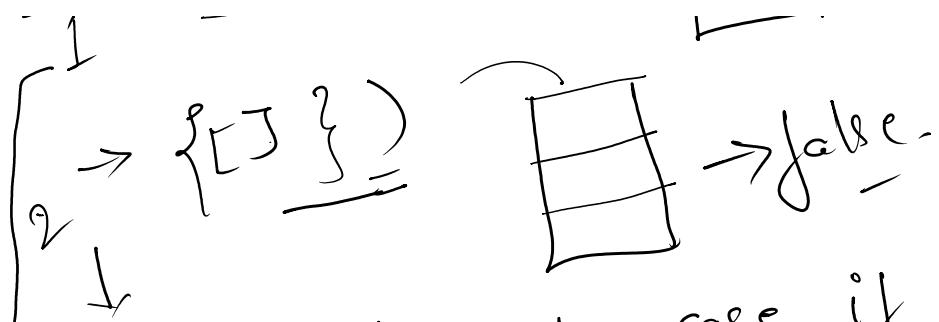


[J ✓
{) ✗

Edge case
1 \rightarrow

({ [] })

\rightarrow false -



To handle this edge case, if closing character is remaining and stack is empty, we will return false.

If all the characters are traversed but still there is character remaining in stack then we will return false.

Code:-

```
Scanner sc = new Scanner(System.in);
String s = sc.next();
Stack<Character> st = new Stack<>();
```

```
for (int i=0; i < s.length(); i++) {
    if (s.charAt(i) == '(' || s.charAt(i) == '{' ||
        s.charAt(i) == '[') {
        st.push(s.charAt(i));
    } else {
        if (st.isEmpty()) {
            . . .
        }
    }
}
```

```
if (st.isEmpty())) {
    . . .
}
```

M2 DAY26 Stack-2 Page 3

```

if (st.isEmpty() || s.charAt(i) == ']' && s.charAt(i+1) == ' ')
    s.o.println("false");
return;

} else {
    char ob = st.pop();
    if (ob == '[' && s.charAt(i) == ']' ||
        ob == '{' && s.charAt(i) == '}' ||
        ob == '(' && s.charAt(i) == ')')
        continue;
    } else {
        s.o.println("false");
        return;
    }

}

s.o.println(st.isEmpty()));
}

```

Postfix Expression Calculation

str = "4572+-*"

$$4 * (5 - (7+2))$$

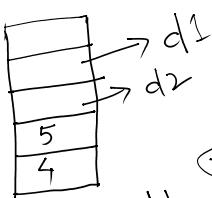
$$4 * (5 - 9) = 4 * -4 = -16$$

Input 2:

str = 964+-

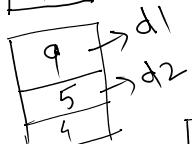
$$9 - (6+4) = 9 - 10 = -1.$$

4572+-*

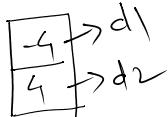


$$d_2 + d_1 = 7 + 2 = 9$$

①



$$d_2 - d_1 = 5 - 9 = -4$$



$$d_2 * d_1 = 4 * -4 = -16$$



st.peek() → is my answer

Code:-

```
Stack<Integer> st = new Stack<>();
for(int i=0; s.length(); i++) {
    if(s.charAt(i) - '0' >= 0 & & s.charAt(i) - '0' <= 9) {
        st.push(s.charAt(i) - '0');
    } else {
        int d1 = st.pop();
        int d2 = st.pop();
        if(s.charAt(i) == '+') {
            st.push(d2 + d1);
        } else if(s.charAt(i) == '-') {
            st.push(d2 - d1);
        } else if(s.charAt(i) == '*') {
            st.push(d2 * d1);
        }
    }
}
```

$$\begin{array}{c} 1234+-* \\ \hline - - - = \\ * (2 - (3 + 4)) \\ \hline \end{array}$$

```

int d2 = s[i];
if (s.charAt(i) == '+') {
    st.push(d2 + d1);
} else if (s.charAt(i) == '-') {
    st.push(d2 - d1);
} else if (s.charAt(i) == '*') {
    st.push(d2 * d1);
} else {
    st.push(d2 / d1);
}
}
s.o.println(st.peek());

```

$$1 \times (-5) = -5$$


$$\begin{array}{l} d1=4 \\ d2=3 \end{array} \quad = 3+4=7$$

$$\begin{array}{l} d1=7 \\ d2=2 \end{array} \quad = 7-2=5$$

$$\begin{array}{l} d1=-5 \\ d2=1 \end{array} \quad = -5 \times 1 = -5$$

Next smaller element to the Right

$$n=9$$

$$Q_{82} = [2, 5, 9, \underline{3}, 1, 12, 6, 8, 7]$$

Output

$$[1, 3, 3, 1, -1, 6, -1, 7, -1]$$

$$n=5$$

$$Q_{82} = [100, 2, 4, 9, 60]$$

Output

$$[2, -1, -1, -1, -1]$$

0 1 2 3

$$Q_{82} = [100, \underline{4}, 2, 9]$$

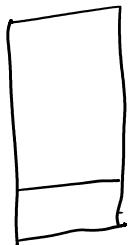
Output

$$[4, 2, -1, -1]$$

Solution:-

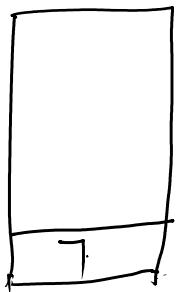
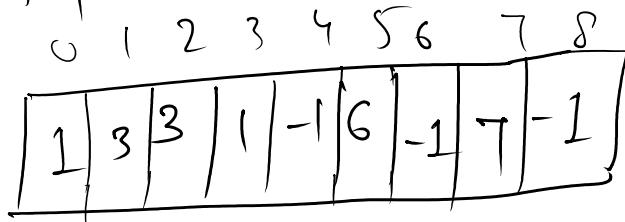
$$\begin{array}{ccccccccccccc} & & & & & & & & & & & & \\ \hline 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & & & & & \\ \hline [2, 5, 9, \underline{3}, 1, 12, 6, 8, \underline{7}] & & & & & & & & & & & & & \end{array}$$

We will start from rightmost



We will create one result array

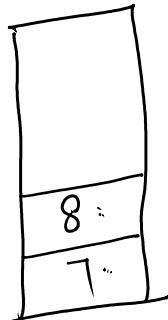
int res[] = new int[n];
index → 8, stack is empty
res[8] = -1



→ we will push 7 into stack

index = 7, value → 8.

res[7] = 7

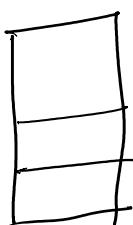


index = 6, value → 6

st.push() = 8 > 6
st.pop()

st.push() = 7 > 6

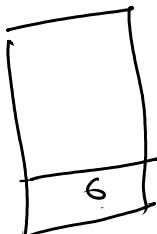
st.pop()



stack is empty()

res[6] = -1

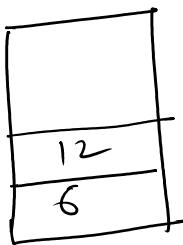
st.push(6)



index = 5, value → 12

st.peek() = 6 < 12

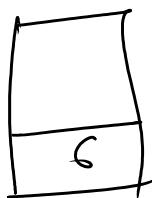
res[5] = 6



index = 4, Value → 1

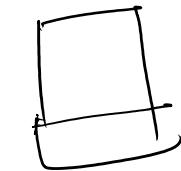
st.peek() = 12 > 1

st.pop();



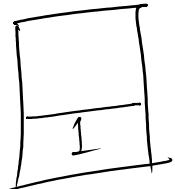
st.peek() = 6 > 1

st.pop();



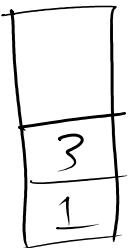
stack is empty

res[4] = -1



index = 3, Value = 3

st.peek() = 1 < 3



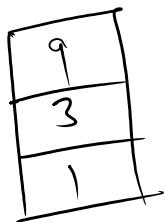
res[3] = 1

st.push(3)

index = 2, Value = 9

st.peek() = 3 < 9

$\text{res}[2] = 3$
 $\text{st.push}(9)$



$\text{index} = 1, \text{Value} = 5$

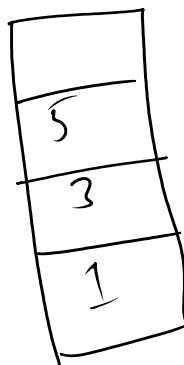
$\text{st.peek}() = 9 > 5$

$\text{st.pop}()$

$\text{st.peek}() = 3 < 5$

$\text{res}[1] = 3$

$\text{st.push}(5)$



$\text{index} = 0, \text{Value} = 2$

$\text{st.peek}() = 5 > 2$

$\text{st.pop}();$

$\text{st.peek}() = 3 > 2$

$\text{st.pop}();$

$\text{st.peek}() = 1 < 2$

$\text{res}[0] = 1$

$\text{st.push}(2)$

Code:-

```
Stack<Integer> st = new Stack<>();
int res[] = new int[n];
for (int i=n-1; i>=0; i--) {
```

```

for (int i=n-1; i>=0; i--) {
    if (st.isEmpty()) {
        yes[i] = -1;
    } else {
        if (st.peek() < arr[i]) {
            yes[i] = st.peek();
            st.push(arr[i]);
        } else {
            while (!st.isEmpty() && st.peek() >= arr[i]) {
                st.pop();
            }
            if (st.isEmpty()) {
                yes[i] = -1;
                st.push(arr[i]);
            } else {
                yes[i] = st.peek();
                st.push(arr[i]);
            }
        }
    }
}

```