Дана скобочная последовательность: [((())()(())]]

В правильной скобочной последовательности количество открытых скобок должно равняться количеству закрытых.

Эта последовательность неверна.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | 5 | 6 |  | 8 |  |  | 11 | 12 | 13 | 14 |
| [ | ( | ( | ( | ) | ) | ( | ) | ( | ( | ) | ) | ] | ] |
| 1 | 2 | 3 | 4 |  |  | 7 |  | 9 | 10 |  |  |  |  |

Формируем стек и записываем в него открывающиеся скобки

|  |  |  |  |
| --- | --- | --- | --- |
| 10 | ( | ) | 11 |
| 9 | ( | ) | 12 |
| 7 | ( | ) | 8 |
| 4 | ( | ) | 5 |
| 3 | ( | ) | 6 |
| 2 | ( | ] | 13 |
| 1 | [ |  |  |

Проходим по строчке из скобок заполняя стек открывающимися скобками до того момента пока нам не попадется закрывающееся (позиция 4 и 5). Мы сравниваем первую закрывающееся скобку с последней добавленной в стек (напоминает принцип работы такой коллекции в языке Java как Queue и ее принцип **Last in first out** ). Если скобки зеркальные мы очищаем строчку нашего стека от открытой скобочки. Дальше поступаем по тому же принципе пока:

1. Не закончится стек (если стек пуст, а в строчке у нас есть еще закрывающиеся строчки – неправильная последовательность).
2. Закончилась строчка, а стек не пуст (в стеке остались данные, а строчка уже закончилась - неправильная последовательность).
3. Верхняя скобка из стека не совпадает с закрывающейся скобкой (наш случай позиция 2 и 13 - неправильная последовательность)

Для приведения нашей последовательности в правильный вид необходима позицию 13 (квадратная закрывающаяся скобка) заменить на круглую закрывающуюся скобку.