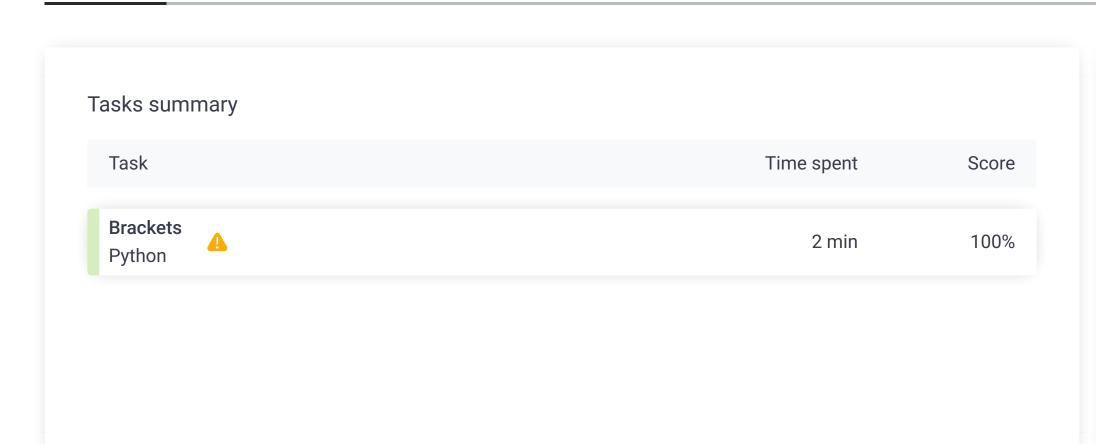
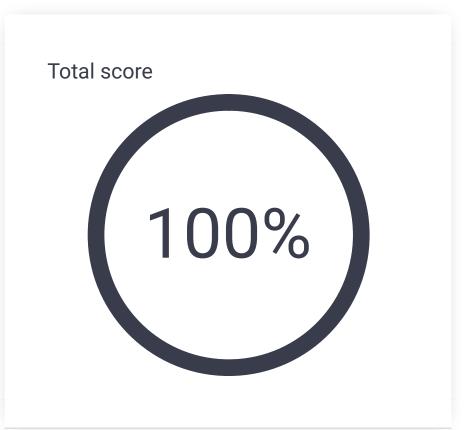
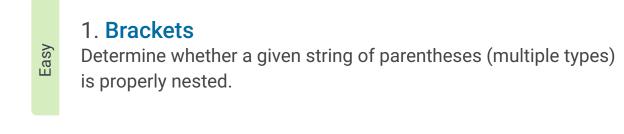
Test Name:

Timeline Summary





Tasks Details



Task Score Performance Correctness 100% 100% 100%

Task description

A string S consisting of N characters is considered to be properly nested if any of the following conditions is true:

- S has the form "(U)" or "[U]" or "{U}" where U is a properly nested
- S has the form "VW" where V and W are properly nested strings.

For example, the string "{ [()()]}" is properly nested but "([)()]" is not. Write a function:

def solution(S)

that, given a string S consisting of N characters, returns 1 if S is properly nested and 0 otherwise.

For example, given $S = "\{[()()]\}$ ", the function should return 1 and given S = "([)()]", the function should return 0, as explained above.

Write an **efficient** algorithm for the following assumptions:

- N is an integer within the range [0..200,000];
- string S consists only of the following characters: "(", "{", "[", "]", "}" and/or ") ".

Copyright 2009–2022 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.

Solution

Solution		
Programming language used:	Python	
Total time used:	2 minutes	9
Effective time used:	2 minutes	•
Notes:	not defined yet	

Task timeline



Code: 19:04:35 UTC, py, final, score: show code in pop-up # you can write to stdout for debugging purposes, e.g. # print("this is a debug message") def solution(S): # write your code in Python 3.6 if len(S) == 0: return 1 if len(S) == 1 or len(S)%2 != 0: 9 return 0 if not $((S[0] == '(' \text{ or } S[0] == '[' \text{ or } S[0] == '\{') \text{ and } (S[0] == '[']))$ 10 11 return 0 12 13 $stack_ = []$ 14 for c in S: 15 if stack_: el = stack_.pop() 16 17 if el == '(' and c == '(': 18 stack_.append(el) stack_.append(c) 19 20 elif el == '(' and c == ')': 21 pass 22 elif el == '(' and c == '[': 23 stack_.append(el) 24 stack_.append(c) 25 elif el == '(' and c == ']': 26 stack_.append(el) 27 stack_.append(c) elif el == '(' and c == '{': 28 29 stack_.append(el) 30 stack_.append(c) elif el == '(' and c == '}': 31 32 stack_.append(el) 33 stack_.append(c) elif el == '[' and c == '(': 34 35 stack_.append(el) 36 stack_.append(c) 37 elif el == '[' and c == ')': 38 stack_.append(el) 39 stack_.append(c) elif el == '[' and c == '[': 40 41 stack_.append(el) 42 stack_.append(c) elif el == '[' and c == ']': 43 44 pass 45 elif el == '[' and c == '{': 46 stack_.append(el) 47 stack_.append(c) elif el == '[' and c == '}': 48 stack_.append(el) 49 50 stack_.append(c) elif el == '{' and c == '(': 51 52 stack_.append(el) 53 stack_.append(c) 54 elif el == '{' and c == ')': 55 stack_.append(el) 56 stack_.append(c) elif el == '{' and c == '[': 57 58 stack_.append(el) 59 stack_.append(c) 60 elif el == '{' and c == ']': stack_.append(el) 61 62 stack_.append(c) elif el == '{' and c == '{': 63 stack_.append(el) 64 65 stack_.append(c) elif el == '{' and c == '}': 66 67 pass

Analysis summary

The solution obtained perfect score.

else:

else:

return 0

return 1

if stack_:

else:

stack_.append(el) stack_.append(c)

stack_.append(c)

Analysis

68

69

70 71

72

73 74

75

76

77

