Codility_

CodeCheck Report: training72AE76-5HE

Test Name:

Check out Codility training tasks

Summary Timeline AI Assistant Transcript

Tasks summary

Task	Time spent	Score
ThreeLetters C#	2 min	100%

Total score

100%

Tasks Details

Medium

1. ThreeLetters

Given two integers A and B, return a string which contains A letters "a" and B letters "b" with no three consecutive letters being the same.

Task Score	Correctness	Performance
100%	100%	Not assessed

Task description

Write a function `solution` that, given two integers A and B, returns a string containing exactly A letters 'a' and exactly B letters 'b' with no three consecutive letters being the same (in other words, neither "aaa" nor "bbb" may occur in the returned string).

Examples:

1. Given A = 5 and B = 3, your function may return "aabaabab". Note that "abaabbaa" would also be a correct answer. Your function may return any correct answer.

2. Given A = 3 and B = 3, your function should return "ababab", "aababb", "abaabb" or any of several other strings.

3. Given A = 1 and B = 4, your function should return "bbabb", which is the only correct answer in this case.

Solution

Programming language used:	C#
Total time used:	2 minutes
Effective time used:	2 minutes
Notes:	*not defined yet*

Task timeline

Assume that:

- A and B are integers within the range [0..100];
- at least one solution exists for the given A and B.

In your solution, focus on **correctness**. The performance of your solution will not be the focus of the assessment.

02:19:05 02:20:18

Code: 02:20:18 UTC, cs, final, score: **100** show code in pop-up

```
using System;
using System.Text;
// you can also use other imports, for example:
// using System.Collections.Generic;

// you can write to stdout for debugging purposes,
// Console.WriteLine("this is a debug message");

class Solution {
    public string solution(int A, int B) {
        // Implement your solution here
        StringBuilder stringBuilder = new StringBui
        int remainingAs = A;
        int remainingBs = B;
        int consecutiveAsCounter = 0;
        int consecutiveBsCounter = 0;

        for (int i = 0; i < A+B; i++)
        {
            if(remainingAs > remainingBs && consecu
            {
                stringBuilder.Append("a");
                remainingAs--;
                consecutiveAsCounter++;
                consecutiveBsCounter = 0;
                continue;
            }

            if (remainingBs > remainingAs && consec
            {
                stringBuilder.Append("b");
                remainingBs--;
                consecutiveBsCounter++;
                consecutiveAsCounter = 0;
                continue;
            }

            if (consecutiveBsCounter == 2)
            {
                stringBuilder.Append("a");
                remainingAs--;
                consecutiveAsCounter++;
                consecutiveBsCounter = 0;
                continue;
            }

            if (consecutiveAsCounter == 2)
            {
                stringBuilder.Append("b");
                remainingBs--;
                consecutiveBsCounter++;
                consecutiveAsCounter = 0;
                continue;
            }

            if (consecutiveAsCounter >= consecutive
            {
                stringBuilder.Append("a");
                remainingAs--;
                consecutiveAsCounter++;
                consecutiveBsCounter = 0;
                continue;
            }
            else
            {
                stringBuilder.Append("b");
                remainingBs--;
```

```
68                  consecutiveBsCounter++;
69                  consecutiveAsCounter = 0;
70                  continue;
71              }
72          }
73
74          return stringBuilder.ToString();
75      }
76  }
```

Analysis summary

The solution obtained perfect score.

Analysis

expand all	Example tests	
▶	example1 first example from the problem statement	✓ OK
▶	example2 second example from the problem statement	✓ OK
▶	example3 third example from the problem statement	✓ OK
expand all	Correctness tests	
▶	zero A == 0 or B == 0	✓ OK
▶	simple simple and very small tests	✓ OK
▶	a_equals_b A == B	✓ OK
▶	a_almost_equals_b |A - B| <= 2	✓ OK
▶	a_greater_than_b A > B	✓ OK
▶	b_greater_than_a B > A	✓ OK
▶	almost_only_one_solution there are very few correct solutions	✓ OK
▶	only_one_solution there is only one correct solution	✓ OK
▶	max A = 100, B = 100	✓ OK
▶	combinations all combinations such that min(A, B) < 3	✓ OK