Repeated Substring Pattern

Given a string `s`, check if it can be constructed by taking a substring of it and appending multiple copies of the substring together.

Example 1:

```
Input: s = "abab"

Output: true

Explanation: It is the substring "ab" twice.
```

Example 2:

```
Input: s = "aba"

Output: false
```

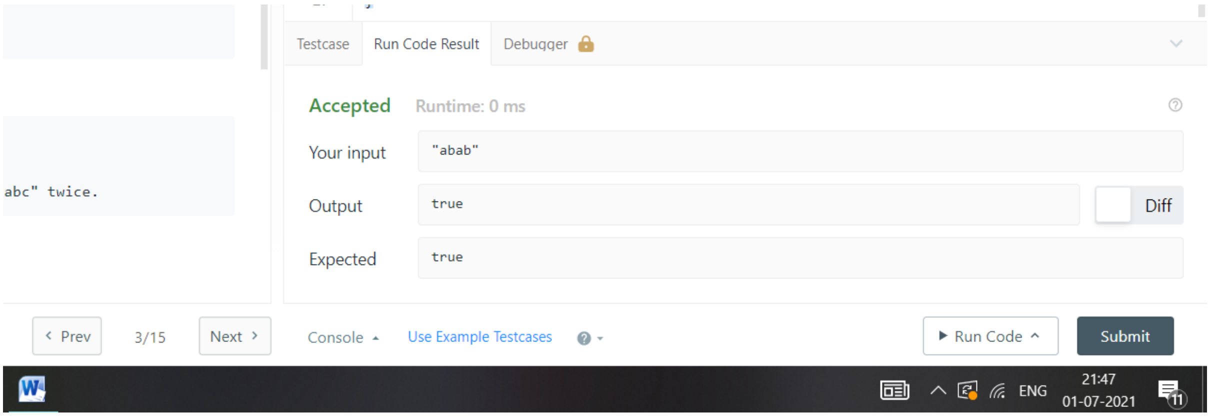
Example 3:

```
Input: s = "abcabcabcabc"

Output: true

Explanation: It is the substring "abc" four times or the substring "abcabc" twice.
```

SOLUTION

```
class Solution {
    public boolean repeatedSubstringPattern(String s) {
        int n = s.length();
        int[] dp = new int[n];
        // Construct partial match table (lookup table).
        // It stores the length of the proper prefix that is also a proper suffix.
        // ex. ababa --> [0, 0, 1, 2, 1]
        // ab --> the length of common prefix / suffix = 0
        // aba --> the length of common prefix / suffix = 1
        // abab --> the length of common prefix / suffix = 2
        // ababa --> the length of common prefix / suffix = 1
        for (int i = 1; i < n; ++i) {
            int j = dp[i - 1];
            while (j > 0 && s.charAt(i) != s.charAt(j)) {
                j = dp[j - 1];
            }
            if (s.charAt(i) == s.charAt(j)) {
                ++j;
            }
            dp[i] = j;
        }

        int l = dp[n - 1];
        // check if it's repeated pattern st
        return l != 0 && n % (n - l) == 0;
    }
}
```

OUTPUT



abc" twice.

Accepted Runtime: 0 ms

Your input "abab"

Output true

Expected true