Goldman Sachs Online Assessment | 01h : 06m : 40s to test end | 1/2 Attempted

☆ Non repeating digit in product

Given a number 'x', and a range of ‘y’ to ‘z’, please find the count of all the numbers ‘n’ in that range, such that the product of the number ‘n’ and 'x' does not contain any digit from the number ‘n’.

General Constraints: *A*ll the inputs will be integers and below 10^5

E.g., if x = 2, y = 10 and z = 15, then,

2 x 10 = 20 // Invalid, since the product contains 0 from 10.

2 x 11 = 22 // Valid

2 x 12 = 24 // Invalid, since the product contains 2 from 12.

2 x 13 = 26 // Valid

2 x 14 = 28 // Valid

2 x 15 = 30 // Valid

Hence, the final count is 4.

Example Input:

2

10

15

Example Output:

4

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour. Start tour ×

Draft saved 12:47 am Original code Java 8

```
1 ▸ import ↔;
```

```java
/*
 * Complete the nonRepeatingDigitProductCount function below.
 */
static int nonRepeatingDigitProductCount(int x, int y, int z) {
    Set<Character> set1 = new HashSet<Character>();
    Set<Character> set2 = new HashSet<Character>();
    if (y > z || x == 1)
        return 0;

    int countInValid = 0;
    for (int i = y; i <= z; i++) {

        long product = i * x;

        for (char ch : (i + "").toCharArray())
            set1.add(ch);

        for (char ch : (product + "").toCharArray())
            set2.add(ch);

        int set2Size = set2.size();
        set2.addAll(set1);
        if (set1.size() + set2Size > set2.size()) {
            countInValid++;
        }
        set1.clear();
        set2.clear();
    }
    return (z - y) - countInValid + 1;
}



private static final Scanner scan = new Scanner(System.in);

public static void main(String[] args) throws IOException {
    BufferedWriter bw = new BufferedWriter(new FileWriter(System.getenv("OUTPUT_PATH")));

    int x = Integer.parseInt(scan.nextLine().trim());

    int y = Integer.parseInt(scan.nextLine().trim());

```

Goldman Sachs Online Assessment | 01h : 06m : 40s to test end | 1/2 Attempted | Smart Coder

```
56          bw.write(String.valueOf(res));
57          bw.newLine();
58
59          bw.close();
60      }
61  }
62
```

Line: 40 Col: 1

Test against custom input

Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

Compiled successfully. All available test cases passed!

Tip: Debug your code against custom input

Test Case #1:	✔	Test Case #3:	✔	Test Case #5:	✔
Test Case #2:	✔	Test Case #4:	✔	Test Case #6:	✔

Testcase 1: ***Success***

Input [Download]

```
2
10
15
```

Your Output

```
4
```

Expected Output [Download]

```
4
```

```
Output hidden
```

Testcase 3: *Success*

Your Output

```
Output hidden
```

Testcase 4: *Success*

Your Output

```
Output hidden
```

Testcase 5: *Success*

Your Output

```
Output hidden
```

Testcase 6: *Success*

Your Output

```
Output hidden
```