Rishabh recently joined a new company and on his very first day his boss gave him a problem to solve.

So, Rishabh don't want to underperform so he asks for you help.

The problem is that you are given with two integers A and B and you have to tell the count of special numbers in range A, B.

A number is called special if the sum of pairwise product of its digits is prime.

For example: suppose abc is a number then its sum of pairwise product of its digits will be (a*b + ac +* b*c).*

Problem Constraints

1 <= A <= B <= 10^9

Input Format

First argument is an integer A.

Second argument is an integer B.

Output Format

Return a single integer denoting the count of special numbers in range [A, B].

Example Input

Input 1:

A = 1

B = 20

Input 2:

A = 100

B = 105

Code:

```
#include "solution.h"
```

```
#include <iostream>
#include <vector>
#include <cmath>
using namespace std;

bool isPrime(int n) {
    if (n <= 1) return false;
    for (int i = 2; i <= sqrt(n); i++) {
        if (n % i == 0) return false;
    }
    return true;
}

int sumOfPairwiseProducts(int n) {
    vector<int> digits;

    while (n > 0) {
        digits.push_back(n % 10);
        n /= 10;
    }

    int sum = 0;

    int size = digits.size();
    for (int i = 0; i < size; i++) {
        for (int j = i + 1; j < size; j++) {
            sum += digits[i] * digits[j];
        }
    }

    return sum;
}

int Solution::countingSpecialNumbers(int A, int B) {
    int count = 0;

    for (int i = A; i <= B; i++) {
        int sum = sumOfPairwiseProducts(i);
        if (isPrime(sum)) {
            count++;
        }
    }

    return count;
}
```