#include <bits/stdc++.h>

using namespace std;

bool isPrime(int n)

{

    // Corner cases

    if (n <= 1)  return false;

    if (n <= 3)  return true;

    // This is checked so that we can skip

    // middle five numbers in below loop

    if (n%2 == 0 || n%3 == 0) return false;

    for (int i=5; i\*i<=n; i=i+6)

        if (n%i == 0 || n%(i+2) == 0)

           return false;

    return true;

}

// count number, that contains prime number of set bit

int primeBitsInRange(int l, int r)

{

    // tot\_bit store number of bit in number

    int tot\_bit, count = 0;

    // iterate loop from l to r

    for (int i = l; i <= r; i++) {

        // use predefined function for finding

        // set bit it is return number of set bit

        tot\_bit = \_\_builtin\_popcount(i);

        // check tot\_bit prime or, not

        if (isPrime(tot\_bit))

            count++;

    }

    return count;

}

// Driven Program

int main()

{

    int l = 6, r = 10;

    cout << primeBitsInRange(l, r);

    return 0;

}