1046. Prime Number of Set Bits in Binary Representation

Given two integers L and R, find the count of numbers in the range [L, R] (inclusive) having a prime number of set bits in their binary representation.

(Recall that the number of set bits an integer has is the number of 1s present when written in binary. For example, 21 written in binary is 10101 which has 3 set bits. Also, 1 is not a prime.)

Example

**Example 1:**

Input: L = 6, R = 10

Output: 4

Explanation:

6 -> 110 (2 set bits, 2 is prime)

7 -> 111 (3 set bits, 3 is prime)

9 -> 1001 (2 set bits , 2 is prime)

10->1010 (2 set bits , 2 is prime)

**Example 2:**

Input: L = 10, R = 15

Output: 5

Explanation:

10 -> 1010 (2 set bits, 2 is prime)

11 -> 1011 (3 set bits, 3 is prime)

12 -> 1100 (2 set bits, 2 is prime)

13 -> 1101 (3 set bits, 3 is prime)

14 -> 1110 (3 set bits, 3 is prime)

15 -> 1111 (4 set bits, 4 is not prime)

Notice

1.L, R will be integers L <= R in the range [1, 10^6].  
2.R - L will be at most 10000.

<https://www.lintcode.com/problem/prime-number-of-set-bits-in-binary-representation/description>

1. // C++ program to generate all prime numbers
2. // less than N in O(N)
3. #include<bits/stdc++.h>
4. #include <iostream>
5. #include <stdio.h>
6. #include <set>
8. using namespace std;

11. static int NumUnos(int n) {
13. int cont = 0;
14. while(n > 0) {
15. int dig = n % 2;
16. if(dig == 1) {
17. cont++;
18. }
19. n /= 2;
20. }
21. return cont;
22. }
24. int countPrimeSetBits(int L, int R) {
25. // Write your code here
26. int primos[] ={ 2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97};
27. int size = sizeof(primos) / sizeof(int);
29. set<int> hash;
30. for(int i =0; i<size; i++) {
31. hash.insert(primos[i]);
32. }

35. int cont = 0;
36. for(int i = L; i<= R; i++) {
37. int unos = NumUnos(i);
38. if( hash.find(unos) != hash.end()) {
39. cont ++;
40. }
41. }
43. return cont;
44. }
46. // driver  program to test above function
47. int main()
48. {
49. */\**
50. *int N = 13 ; // Must be less than MAX\_SIZE*
52. *manipulated\_seive(N);*
54. *// pint all prime number less then N*
55. *for (int i=0; i<prime.size() && prime[i] <= N ; i++)*
56. *cout << prime[i] << " ";*
57. *\*/*
58. //int  L = 6, R = 10;
59. int L = 10, R = 15;
61. //int L =
62. cout << countPrimeSetBits( L,  R) << endl;
64. return 0;
65. }