**★★☆☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：100: Goldbach and Euler**

**解題者：林其璜**

**解題日期：2022年3月10日**

**題意：**

給定一個正整數 n (0 < n ≤ 100,000,000)，判斷是否存在兩個質數 p1、p2，使得 p1 + p2 = n，同時要保證 p2 – p1的值最小且大於 0。

要注意：

p1 要小於 p2（不能等於）

此題中 1 也算是質數

**題意範例：**

11 → 11 is not the sum of two primes!

12 → 12 is the sum of 5 and 7.

**解法：**

建立 100,000,000 以下的質數表，如果 n 是奇數，只要查詢 n – 2 是否為質數即可。如果 n 是偶數，由於 p1、p2 要盡可能靠近，因此我們從質數表中透過二分搜尋法找出最靠近且大於等於 n / 2 的質數 p。

接著我們搜尋 n – p 是否在質數表中，如果不在，那就將 p 設成下一個質數，然後再次查詢 n – p，直到找到，或是 n – p 小於 1 為止。

**解法範例：**

無。

**討論：**

(1) 善用 STL 裡的 lower\_bound 跟 binary\_search 函數。

(2) 通常建立質數表是利用刪除自己的倍數的方法來建立的，也就是刪除

2p 3p 4p …，但我們其實不需要從 2p 開始，從 p \* p 開始刪就好了。

**程式：**

#include <iostream>

#include <algorithm>

using namespace std;

#define TAB\_SIZE 100000001

bool prime\_table[TAB\_SIZE] = { 1,1 };

int primes[TAB\_SIZE / 2];

int \_count = 0;

void construct\_table() {

 for (int i = 1; i < TAB\_SIZE; i++) {

 if (prime\_table[i])

 continue;

 if (i < 32768)

 for (int k = i \* i; k < TAB\_SIZE; k += i)

 prime\_table[k] = 1;

 primes[\_count++] = i;

 }

}

int main() {

 construct\_table();

 int n;

 while (cin >> n) {

 if (n % 2) {

 int\* ptr = lower\_bound(primes, primes + \_count, n - 2);

 if (ptr != primes + \_count && \*ptr == n - 2) {

 cout << n << " is the sum of " << 2 << " and " << \*ptr << "." << endl;

 continue;

 }

 }

 else {

 bool flag = false;

 int\* ptr = lower\_bound(primes, primes + \_count, n / 2);

 for (int k; (k = n - \*ptr) >= 2 && ptr != primes + \_count; ptr++)

 if (binary\_search(primes, ptr + 1, k) && k != \*ptr) {

 cout << n << " is the sum of " << k << " and " << \*ptr << "." << endl;

 flag = true;

 break;

 }

 if (flag)

 continue;

 }

 cout << n << " is not the sum of two primes!" << endl;

 }

}