**Primes sum**

Submissions: [6133](https://practice.geeksforgeeks.org/problem_submissions.php?pid=2196)  Accuracy:

13.63%

   Difficulty: [Easy](https://practice.geeksforgeeks.org/Easy/0/0/)   Marks: 2

Show Topic Tags   

[Zoho](https://practice.geeksforgeeks.org/company/Zoho/)

Given a number N. Find if it can be expressed as sum of two prime numbers.

**Input:**  
The first line of input contains an integer T denoting the number of test cases. Then T test cases follow. Each test case contains an integer N as input.

**Output:**  
For each test case, In new line print "Yes" if it can be expressed, Otherwise print "No".

**Constraints:**  
1<=T<=2000  
1<=N<=106

**Example:**  
**Input:**  
2  
34  
23

**Output:**  
Yes  
No

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/primes-sum/0/?ref=self#ExpectOP) option \*\*

Contributor: Arun Tyagi  
[Author: arun03](https://auth.geeksforgeeks.org/user/arun03/practice/)

<https://practice.geeksforgeeks.org/problems/primes-sum/0/?ref=self>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp1

{

class Program

{

// function generate all prime number less then N in O(n)

static HashSet<int> manipulated\_seive(int N)

{

// 0 and 1 are not prime

// isprime[0] = isprime[1] = false;

List<bool> isprime = new List<bool>(N + 1);

List<int> prime = new List<int>(N + 1);

List<int> SPF = new List<int>(N + 1);

isprime.Add(false);

isprime.Add(false);

SPF.Add(0);

SPF.Add(0);

for (int i = 2; i < N + 1; i++)

{

isprime.Add(true);

SPF.Add(0);

}

// Fill rest of the entries

for (int i = 2; i < N; i++)

{

// If isPrime[i] == True then i is

// prime number

if (isprime[i])

{

// put i into prime[] vector

prime.Add(i);

// A prime number is its own smallest

// prime factor

SPF[i] = i;

}

// Remove all multiples of i\*prime[j] which are

// not prime by making isPrime[i\*prime[j]] = false

// and put smallest prime factor of i\*Prime[j] as prime[j]

// [ for exp :let i = 5 , j = 0 , prime[j] = 2 [ i\*prime[j] = 10 ]

// so smallest prime factor of '10' is '2' that is prime[j] ]

// this loop run only one time for number which are not prime

for (int j = 0;

j < (int)prime.Count &&

i \* prime[j] < N && prime[j] <= SPF[i];

j++)

{

isprime[i \* prime[j]] = false;

// put smallest prime factor of i\*prime[j]

SPF[i \* prime[j]] = prime[j];

}

}

return new HashSet<int> ( prime );

}

static bool EsSuma2(int N, HashSet<int> primos)

{

int[] arr = primos.ToArray();

for(int i =0; i<arr.Length; i++)

{

if(primos.Contains(N - arr[i]))

{

return true;

}

}

return false;

}

static void Main(string[] args)

{

HashSet<int> primos = manipulated\_seive(1000000);

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

int N = int.Parse(Console.ReadLine());

Console.WriteLine(EsSuma2(N, primos) ? "Yes" : "No");

}

Console.ReadLine();

}

}

}