## 1. Strong Prime

In number theory, **Strong Prime** is a prime number **P** which is greater than the arithmetic mean of **PP** and **NP**.

Where

- **PP** is previous_prime of **P**.
- **NP** is next_prime of **P**.

**Examples:**
P = 17 is a **Strong Prime** as it is greater than arithmetic mean of PP = 13 and NP = 19, which is (13 + 19) / 2 = 32 / 2 = 16.

P = 29 is a **Strong Prime** as it is greater than arithmetic mean of PP = 25 and NP = 31, which is (25 + 31) / 2 = 56 / 2 = 28.

P = 35 is **Not a Strong Prime** as though it is greater than arithmetic mean of PP = 31 and NP = 37, which is (31 + 37) / 2 = 68 / 2 = 34. **35 is not a Prime Number itself.**

Given a number **N** determine if it's **Strong Prime or not**.

**Input Format:**
The only line of input contains a single number **N**.

**Output Format:**
Print **YES** if **N** is Strong Prime,
**NO** otherwise

**Constraints:**
1 <= N <= 1000000

**Sample I/O:**

**Input 1:**
11
**Output 1:**
YES

**Input 2:**
17
**Output 2:**
YES

**Input 3:**
13
**Output 3:**
NO

**Input 4:**
7
**Output 4:**
NO

## 2. Kohli and Coins

Kohli has infinite coins in denominations of rupees 5 and rupees 10.
Find the minimum number of coins Kohli needs, to pay exactly X rupees. If it is impossible to pay X rupees in denominations of rupees 5 and 10 only, print -1.

**Sample I/O:**

**Input Format:**
A single line contains an integer X.

**Output Format:**
Print the output according to the description.

**Input 1:**
50

**Output 1:**
5

**Input 2:**
15

**Output 2:**
2

**Input3:**
8

**Output 3:**
-1

**Input 4:**
95

**Output 4:**
10