Problem C
Canonical Coin Systems

Image by Diane Wiens, Used with permission

A *coin system* S is a finite (nonempty) set of distinct positive integers corresponding to coin values, also called *denominations*, in a real or imagined monetary system. For example, the coin system in common use in Canada is $\{1, 5, 10, 25, 100, 200\}$, where 1 corresponds to a 1-cent coin and 200 corresponds to a 200-cent (2-dollar) coin. For any coin system S, we assume that there is an unlimited supply of coins of each denomination, and we also assume that S contains 1, since this guarantees that any positive integer can be written as a sum of (possibly repeated) values in S.

Cashiers all over the world face (and solve) the following problem: For a given coin system and a positive integer amount owed to a customer, what is the smallest number of coins required to dispense exactly that amount? For example, suppose a cashier in Canada owes a customer 83 cents. One possible solution is 25+25+10+10+10+1+1+1, i.e., 8 coins, but this is not optimal, since the cashier could instead dispense $25 + 25 + 25 + 5 + 1 + 1 + 1$, i.e., 7 coins (which *is* optimal in this case). Fortunately, the Canadian coin system has the nice property that the *greedy algorithm* always yields an optimal solution, as do the coin systems used in most countries. The greedy algorithm involves repeatedly choosing a coin of the largest denomination that is less than or equal to the amount still owed, until the amount owed reaches zero. A coin system for which the greedy algorithm is always optimal is called *canonical*.

Your challenge is this: Given a coin system $S = \{c_1, c_2, \ldots, c_n\}$, determine whether S is canonical or non-canonical. Note that if S is non-canonical then there exists at least one *counterexample*, i.e., a positive integer x such that the minimum number of coins required to dispense exactly x is less than the number of coins used by the greedy algorithm. An example of a non-canonical coin system is $\{1, 3, 4\}$, for which 6 is a counterexample, since the greedy algorithm yields $4 + 1 + 1$ (3 coins), but an optimal solution is $3 + 3$ (2 coins). A useful fact (due to Dexter Kozen and Shmuel Zaks) is that if S is non-canonical, then the smallest counterexample is less than the sum of the two largest denominations.

Input

Input consists of a single case. The first line contains an integer n $(2 \le n \le 100)$, the number of denominations in the coin system. The next line contains the n denominations as space-separated integers $c_1\ c_2\ \ldots\ c_n$, where $c_1 = 1$ and $c_1 < c_2 < \ldots < c_n \le 10^6$.

Output

Output "`canonical`" if the coin system is canonical, or "`non-canonical`" if the coin system is non-canonical.

Sample Input 1	Sample Output 1
`4` `1 2 4 8`	`canonical`

Sample Input 2	Sample Output 2
3 1 5 8	non-canonical

Sample Input 3	Sample Output 3
6 1 5 10 25 100 200	canonical