Problem A. Mancala-esque Mayhem

Input file: standard input
Output file: standard output
Time & memory limit: N/A (Better time, memory, or complexity is scored)

Minded is developing an autoscorer for Mancala and needs to determine whether, played optimally, a player can capture all the marbles on their side of the board without any overflowing to the opponent's side.

The board state is given as 9 integers. The first value is the player's capture pile (index 0), and the remaining 8 values are the marble counts in each of the player's pits (indices $1 \rightarrow 8$). The sum of all marbles, S_m, does not exceed 48.

Input

A single line containing 9 space-separated integers. Their sum S_m satisfies $S_m \leq 48$.

Output

Print "YES" or "NO" indicating whether all marbles can be captured without overflow, then print the maximum number of marbles that can be captured without overflowing into the opponent's side followed by the final board state.

Examples

Standard input	Standard output
0 1 2 3 4 5 6 7 8	YES 36 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 9	NO 28 0 0 0 0 0 0 0 9
0 2 0 0 0 0 0 0 0	NO 0 2 0 0 0 0 0 0 0
0 1 3 3 4 5 6 7 8	NO 23 0 14 0 0 0 0 0 0

Note

This isn't played like a standard Mancala round; there are no switching turns, only the aforementioned rules apply.

Problem B. Enemy Base

Input file: standard input
Output file: standard output
Time & memory limit: N/A (Better time, memory, or complexity is scored)

After days of reconnaissance, the joint strike force is finally closing in on the enemy's hidden military base. Enemy fighter jets patrol the area but must periodically return to base for refueling. Each time a jet returns, the long-range radar installation records its position, but atmospheric interference and radar calibration limits mean these readings aren't perfectly accurate. Over the course of the operation, radar records n jet returns $(1 \leq n \leq 10^5)$. For the i-th return, the system records:

- x_i, y_i $(-10^6 \leq x_i, y_i \leq 10^6)$: Measured position of the enemy base.
- a_i $(0 \leq a_i \leq 10^6)$: Accuracy of the fighter jet pilot.

The command team needs to estimate the base's true position (b_x, b_y) by finding the value that best fits all observed jet positions. The targeting algorithm works by finding the position that minimizes this error function:

$$\sum_{i=1}^{n} a_i \cdot \sqrt{(b_x - x_i)^2 + (b_y - y_i)^2}$$

Help the command team determine the base's most likely location.

Input

The first line contains a single integer n $(1 \le n \le 10^5)$, the number of jet returns recorded.
The next n lines each contain three integers x_i, y_i, and a_i, representing the measured position and radar accuracy rating of the i-th return.

Output

Output two numbers b_x and b_y, the base's estimated position on the 2D plane that minimizes the error function. If multiple positions achieve the minimum, output any one of them. Your answer will be accepted if the absolute or relative error is less than 10^{-6}.

Examples

Standard input	Standard output
1 0 0 28	0.00000000 0.00000000
0 1 2 3 4 5 6 7 9	NO 28 0 0 0 0 0 0 0 9
0 2 0 0 0 0 0 0 0	NO 0 2 0 0 0 0 0 0 0
0 1 3 3 4 5 6 7 8	NO 23 0 14 0 0 0 0 0 0

Problem C. Mod Problem

Input file: standard input
Output file: standard output
Time & memory limit: N/A (Better time, memory, or complexity is scored)

In the land of Arithmetica, a young mage starts with the number 1 and wants to transform it into a given target N using three ancient spells. Each spell changes the current number as follows:

- Spell of Doubling Power: multiply by 2^P
- Spell of the Prime's Kin: multiply by $P + 1$
- Spell of Squaring: square the number

After casting a spell, the result is immediately reduced modulo P^2 (where P is a prime). The mage can cast any spell any number of times, in any order.

Your quest is to determine the minimum number of spell casts needed to reach N, or report that it is impossible.

Input

The first line contains an integer T $(1 \le T \le 10^3)$, the number of test cases.
Each of the next T lines contain two integers: P $(3 \le P \le 5 \cdot 10^4$ and N $(1 \le N < P^2)$.

It is guaranteed that the sum of all P over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, output a single integer: the minimal number of spells, or -1 if the target cannot be obtained.

Examples

Standard input	Standard output
3 3 1 3 4 3 2	0 1 3

Note

For the first case, already at $1 \rightarrow 0$ spells.

For the second case: $1 \cdot (P + 1) \rightarrow 4 \ (mod\ 9) \rightarrow 1$ spell.

For the third case: $1 \cdot 4 \rightarrow 4^2 \rightarrow 16 \equiv 7 \ (mod\ 9) \cdot 2^P \rightarrow 56 \equiv 2 \ (mod\ 9) = 3$ spells. It can be shown that fewer than 3 spells cannot produce 2.