Number Game

(from 2001 North-Western Europe Regional Contest)

`game.p`, `game.c`, `game.C`

Christiane and Matthias are playing a new game, the *Number Game*. The rules of the Number Game are: Christian and Matthias take turns in choosing integer numbers greater than or equal to 2. The following rules restrict the set of numbers which may be chosen:

R1: A number which has already been chosen by one of the players or a multiple of such a number cannot be chosen. (A number z is a *multiple* of a number y if z can be written as $y.x$ and x is a positive integer.)

R2: A sum of two such multiples cannot be chosen either.

R3: For simplicity, a number which is greater than 20 cannot be chosen either. This enables a lot more NPCs (Non-Personal-Computers) to play this game.

The player who cannot choose any number anymore looses the Number Game.

Here is an example: Matthias starts by choosing 4. Then Christiane is not allowed to choose 4, 8, 12, etc. Let us assume her move is 3. Now, the numbers 3, 6, 9, etc. are excluded, too; furthermore, numbers like: $7 = 3 + 4$, $10 = 2.3 + 4$, $11 = 3 + 2.4$, $13 = 3.3 + 4$, ... are not available. So, in fact, the only numbers left are 2 and 5. Matthias now says 2. Since $5 = 2 + 3$ is now forbidden, too, he wins because there is no number for Christiane s move left.

Your task is to write a program which will help to play the Number Game. In general, i.e., without rule R3, this game may go on forever. However, with rule R3, it is possible to write a program that finds a strategy to win the game.

Problem

Given a game situation (a list of numbers which are not yet forbidden), your program should output all *winning moves*. A winning move is a move by which the player whose turn it is can force a win no matter what the other player will do. Now we define these terms more formally:

- A *loosing position* is a position in which either
  1. all numbers are forbidden, or
  2. no winning move exists.
- A *winning position* is a position in which a winning move exists.
- A *winning move* is a move after which the position is a loosing position.

Input Specification

The first line contains the number of scenarios.

The input for each scenario describes a game position. It begins with a line containing the number a, $0 \leq a < 20$ of numbers which are still available. Next follows a single line with the a numbers still available, separated by single blanks.

You may assume that all game positions in the input could really occur in the Number Game (for example, if 3 is not in the list of numbers available, 6 will not be, either).

Output Specification

The output for each scenario begins with a line containing "`Scenario #`i`:`", where i is the number of the scenario starting at 1. In the next line either print "`There is no winning move.`" if this is true for the position of the current scenario, or "`The winning moves are:` w_1 w_2 `...` w_k`.`" where the w_i are all the winning moves, in ascending order, separated by single blanks. The output for each scenario should be followed by a blank line.

Sample Input

```
2
1
2
2
2 3
```

Output for Sample Input

```
Scenario #1:
The winning moves are: 2.

Scenario #2:
There is no winning move.
```