

## Problem E. Eternal Masters

Input file: *standard input*  
 Output file: *standard output*  
 Time limit: 1 second  
 Memory limit: 256 mebibytes

*This is an interactive problem.*

Two players, Red and White, are playing a game according to the following rules (inspired by the rules of “Magic: the Gathering”):

- White has  $L$  life total;  $L$  is a positive integer.
- Red’s goal is to decrease  $L$  to zero or below. White’s goal is to prevent that.
- Red has  $n$  cards in her hand. The  $i$ -th card can decrease  $L$  by a positive integer  $r_i$ .
- White has  $m$  cards in her hand. The  $i$ -th card can increase  $L$  by a positive integer  $w_i$ .
- Each card can only be played from hand at most once.
- Players know each other’s cards.
- Red’s and White’s turns alternate; the first turn is Red’s turn.
- On each turn, the player whose turn it is can either play a card from her hand (if there is any) or pass.
- There is a zone named “stack”, much like a programmer’s stack. Initially the stack is empty. Playing a card from the hand **does not** immediately trigger its effect. Instead, it causes the card to be placed on top of the stack. The stack is shared for both players’ cards.
- A pass causes the top card of the stack (which, as we can show, is always an opponent’s card) to deal its effect. Then the card is removed from the stack and discarded.
- If White’s pass causes  $L$  to become zero or less, White loses immediately.
- If Red passes while the stack is empty, Red loses immediately.
- It can be shown that every game will eventually end in one of the two ways described above.

Given  $L$ ,  $n$ ,  $r$ ,  $m$ , and  $w$ , select a player and play for that player against the interactor to win.

### Interaction Protocol

The interaction starts with reading the given game state.

The first line contains an integer  $n$  ( $1 \leq n \leq 1000$ ): the number of cards in Red’s hand. The next  $n$  lines contain  $n$  integers  $r_1, r_2, \dots, r_n$  ( $1 \leq r_i \leq 10^6$ ): the values of Red’s cards, one value per line.

The next line contains an integer  $m$  ( $1 \leq m \leq 1000$ ): the number of cards in White’s hand. The next  $m$  lines contain  $m$  integers  $w_1, w_2, \dots, w_m$  ( $1 \leq w_i \leq 10^6$ ): the values of White’s cards, one value per line.

Different cards may have the same value.

The next line contains an integer  $L$  ( $1 \leq L \leq 10^6$ ): the initial life total.

After reading all these values, print a single line containing either a word “Red” or “White”. This denotes the player you will play for.

After that, the game starts with Red’s turn.

At your turn, if you want to pass, print “pass”. If you want to play a card from your hand, print “play  $x$ ”, where  $x$  is the value of the card you want to play; that card must be available to you.

At the interactor's turn, read a line. This line says either "pass" or "play  $x$ ", where  $x$  is the value of the card that the interactor has played from the hand.

If the interactor's pass causes you to win, your program should print "win" and terminate gracefully.

If your pass causes you to lose, the interactor will print "win". After reading that line, terminate your program gracefully to receive the "Wrong Answer" verdict.

After printing something, do not forget to output end of line and flush the output. Otherwise, you will get the "Idleness limit exceeded" verdict. To flush the output this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

## Example

<i>standard input</i>	<i>standard output</i>
3	
6	
2	
2	
1	
9	
6	White
play 2	pass
play 2	pass
play 6	play 9
pass	pass
pass	win