EEE472/CSE422 [C02]
Strange Bank Problem

Suppose, you are the owner of a bank that operates in a strange way. Customers can lend money from your bank (just like a normal bank) and they can also deposit money in your bank. A register is maintained to track the daily transactions. However, being the strange owner of a strange bank, you have a fascination with finding out whether a portion of your daily transactions (in/out) balance out to zero. For example, suppose

1	Lend	100
2	Deposit	150
3	Lend	400
4	Lend	500
5	Deposit	1000
6	Lend	460
7	Deposit	160
8	Deposit	200
9	Lend	500
10	Deposit	100

In this case, there is a portion of the transactions that would balance itself out. (6th, 7th, 8th, and 10th transactions would amount to 0).

Your task is to use a genetic algorithm to solve this strange bank problem.

Task Breakdown:

1. Model the transaction register in a way suitable for the problem.
2. Write a fitness function. Hint: It is the sum of the non-zero elements of a register.
3. Write the crossover function.
4. Write the mutation function.
5. Create a population of randomly generated registers.
6. Run genetic algorithms on the population until highest fitness has been reached and/or number of maximum iterations has been reached.

Input

The first line has a number denoting the number of daily transactions followed by lines each starting with either ***l*** or ***d*** and a number denoting the amount of transaction. Here:

Output

The output would be a binary string denoting the specific transactions that balance themselves to zero or -1 if such a string cannot be formed. String consisting of all zeros won't be accepted.

Example:

Sample Input 1
7 l 120 l 289 d 475 l 195 d 6482 l 160 d 935
Sample Output 1
1011010

Sample Input 2
5 I 100 I 450 d 500 I 7923 d 9055
Sample Output 2
-1