

Fewest Coins Needed to Make Change

Filename: mincoins

The Problem

Given a set of coin denominations, determine the fewest number of coins necessary to make change for a given number of cents. Assume you can use as many of one particular denomination as necessary. For example, if the set of coin denominations was {1, 5, 10, 25} one would need 3 coins to make change for 40 cents: 25, 10 and 5. You will also need to detect cases where it's impossible to make change for the given amount.

The Input

The first line of the input file will contain a single positive integer, n ($n < 100$), representing the number of input cases. The following n lines contain information for one input case each. The first number on each of these lines is a positive integer, x ($x < 1000000$), representing the number of cents for which to make change. The second number on each of these lines will be a positive integer, d ($d < 10$), representing the number of different coin denominations for the input case. This is followed by d distinct positive integers less than 10000, sorted in ascending order, representing the different coin denominations for the input set.

The Output

For each input case, output a single integer on a line by itself representing the fewest number of coins necessary to make change for the given number of cents. If no such way exists, output -1.

Sample Input

```
3
11 4 1 5 10 25
3 9 1 2 3 4 5 6 7 8 9
5 3 2 4 6
```

Sample Output

```
2
1
-1
```

Combinations

Filename: combo

The Problem

Given non-negative integers n and k , where $k \leq n$, calculate the number of ways to choose k items out of n . Since the answer to these queries might be quite high, return your answer mod 10^{18} .

The Input

The first line of the input file will contain a single positive integer, C ($C < 100$), representing the number of input cases. The following C lines contain information for one input case each. On each of these lines there will be two non-negative integers, n and k , separated by a single space, where $0 \leq k \leq n \leq 1000$.

The Output

For each input case, output a single integer on a line by itself representing the number of ways to choose k items out of n for the given input case, mod 10^{18} .

Sample Input

```
2
5 2
10 5
```

Sample Output

```
10
252
```

Number of Ways to Make Change

Filename: numways

The Problem

Given a set of coin denominations, determine the number of ways to make change for a given number of cents. Assume you can use as many of one particular denomination as necessary. For example, if the set of coin denominations was {1, 5, 10, 25}, then there would be four ways to make change for 11 cents, as follows:

10, 1
5, 5, 1
5, 1, 1, 1, 1, 1
1, 1, 1, 1, 1, 1, 1, 1, 1, 1

Since the answer to one of these queries might be quite high, return your answer mod $10^9 + 7$.

The Input

The first line of the input file will contain a single positive integer, n ($n < 100$), representing the number of input cases. The following n lines contain information for one input case each. The first number on each of these lines is a positive integer, x ($x < 10000$), representing the number of cents for which to make change. The second number on each of these lines will be a positive integer, d ($d < 10$), representing the number of different coin denominations for the input case. This is followed by d distinct positive integers less than 10000, sorted in ascending order, representing the different coin denominations for the input set.

The Output

For each input case, output a single integer on a line by itself representing the number of ways to make change for the given input case, mod 10^9+7 .

Sample Input

```
2
11 4 1 5 10 25
3 9 1 2 3 4 5 6 7 8 9
```

Sample Output

```
4
3
```

Chess Battles

Filename: chess

The Problem

Alice and Bob have been planning a long-term chess tournament. In their probability class, they learned that in any competition, if more games are played, it's more likely that the "better" player will win a majority of the games. Thus, there's a higher chance for the "better" player to win in a best of 19 series than a best of 7 series. In addition, the winner of the previous match gets to play with white in the subsequent match and the player that gets to play with white for the first match is the designated first player. Alice and Bob would like for you to write a program that, given the number of total games being played in the series, and the probabilities that the first player given wins with white and wins with black, calculates the probability of the first player winning the series. You may assume that no matches end in draws.

The Input

The first line of the input file will contain a single positive integer, C ($C < 100$), representing the number of input cases. The following C lines contain information for one input case each. The first value on each of these lines will be a positive odd integer, n ($n < 1000$), representing the number of games to be played in the series. The next two values on each line will be two floating point numbers, w and b , representing the probabilities, respectively that the first player wins a given match playing with the white pieces and playing with the black pieces. Both of these values will be in between 0 and 1, inclusive.

The Output

For each input case, output a floating point number rounded to six decimal places representing the probability that the first player wins the given series.

Sample Input

```
2
5 .5 .5
1 .7 .3
```

Sample Output

```
.500000
.700000
```