Day 1 Question 1

Fibonacci Numbers

Input file: fib.in

Output file: fib.out

The nth Fibonacci number, f(n), is defined thus:

f(1) = 1
f(2) = 1
f(n) = f(n-1) + f(n-2) [for all n > 2]

Write a program that reads several n, one per line, and writes the corresponding f(n), one per line. Each value of n will be between 1 and 200. The last line of input contains 0.

Sample input

```
1
2
3
4
5
100
0
```

Output for sample input

```
1
1
2
3
5
354224848179261915075
```

Day 1 Question 2

Message Deciphering

Input file: crypt.in

Output file: crypt.out

Messages can be ciphered (encrypted) by systematically replacing letters of the alphabet by other letters. A simple cipher known as the Caesar cipher replaces each letter in the alphabet by the letter k positions later in the alphabet, where A is considered to follow Z . For example, if k = 6, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, and Z would be replaced by G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, A, B, C, D, E, and F respecively. The message

```
THE FULL MOON RISING IS A BAD SIGN
```

would be ciphered as

```
ZNK LARR SUUT XOYOTM OY G HGJ YOMT
```

The inverse of the cipher is again a Caesar cipher with 26-k replacing k.

Your job as cryptanalist is to decode lines of text that have been encoded with a Ceasar cipher, each using a different unknown value of k. For example, if the input were

```
ZNK LARR SUUT XOYOTM OY G HGJ YOMT
FA NQ AD ZAF FA NQ FTMF UE FTQ CGQEFUAZ
```

the output would be

```
THE FULL MOON RISING IS A BAD SIGN
TO BE OR NOT TO BE THAT IS THE QUESTION
```

(the first line was ciphered with k=6 and the second line with k=12).

Of course there are 26 possible values of k and therefore 26 possible ciphers, so you will have to "guess" by selecting the most probable deciphering. The probability of a particular deciphering can be estimated using the probabilities of the letters it contains. In English, E is the most common letter, with probability 0.127, T is the next more common with probability 0.091, and so on. A complete table of letter probabilities is given below. The probability of a complete line of text can be approximated by the product of the probabilities of the letters it contains.

Input Specification

The input to your program consists of a line containing a positive integer n, followed by n lines each consisting of of upper case letters and spaces only. Each line is an English phrase or sentence, encrypted with a Caesar cipher with unknown k.

Output Specification

For each line of input, give the most probable deciphering.

Sample Input

```
2
ZNK LARR SUUT XOYOTM OY G HGJ YOMT
FA NQ AD ZAF FA NQ FTMF UE FTQ CGQEFUAZ
```

Output for Sample Input

```
THE FULL MOON RISING IS A BAD SIGN
TO BE OR NOT TO BE THAT IS THE QUESTION
```

Probabilities of Letters in English Text

Letter	Probability	Letter	Probability
A	.082	N	.067
B	.015	O	.075
C	.028	P	.019
D	.043	Q	.001
E	.127	R	.060
F	.022	S	.063
G	.020	T	.091
H	.061	U	.028
I	.070	V	.010
J	.002	W	.023
K	.008	X	.001
L	.040	Y	.020
M	.024	Z	.001

Day 1 Question 3

Bus Schedule

Input file: bus.in

Output file: bus.out

A city contains several bus routes. Each route consists of a number of stops and busses periodically travel among these stops. A traveller wishes to make a journey from one stop to another, arriving no later than a specified time. Your job is to determine the latest time the traveller may arrive at the initial stop so as to be able to arrive at the destination on time.

The input consists of a number of schedules. A schedule consists of

- A beginning time (an exact hour between 0 and 24).
- An ending time (an exact hour between 0 and 24).
- A number n indicating the number of stops on the route.
- A list of n stop numbers, one per line (each stop has a unique number between 1 and 1000).
- A list of n-1 time intervals in minutes, one per line.

There are no more than 50 schedules and each schedule has no more than 50 stops. Following the last schedule is a line containing -1.

The busses operate as follows. For each schedule, the bus leaves at the beginning time and visits its stops in order. The time interval between the stops is as indicated in the schedule. At the last stop the bus turns around and revisits the stops in reverse order with the same time intervals. This process repeats until the ending time at which time the bus returns home without visiting any more stops (passengers on the bus at the end time are in serious trouble).

Following the schedules (and the line containing -1) are several traveller requests. Each traveller request consists of

- the stop at which the journey begins
- the stop at which the journey ends
- the latest time at which the journey can end (hours and minutes on two separate lines).

The last traveller request is followed by another line containing -1.

For each traveller request, print the latest time that the traveller can be at the beginning stop and still reach the destination in time. It will always be possible to make the trip.

Sample input

```
6
22
```

```
11
1
2
3
4
5
6
7
8
9
10
11
6
6
6
6
6
6
6
6
6
6
0
24
4
6
16
26
36
20
20
20
-1
1
11
15
0
1
11
14
59
11
36
15
0
-1
```

Output for Sample Input

```
14:00
12:00
13:00
```