

We are going to simulate a competition of cycling against the clock, where each participant leaves every X units of time, and must travel a distance D . The simulation is represented by a string S with at most 300 characters. Each character represents the player's movement at each unit of time, that is, if the character C_i is 1, means that player 1 advanced at moment i ($1 \leq i \leq 300$). For easiness, only one player will advance at a time. Player number 1 leaves at moment 1, player 2 leaves at moment $1 + X$, player 3 leaves at moment $1 + 2X$, and so on. If a player that hasn't left appears, his movement must be ignored. A player finishes the race when appears at least D times after leaving.



For example, the string "122121" with $N = 2$, $X = 2$ and $D = 2$, means that player 1 advances one unit at time 1. The character number two must be ignored because the second player hasn't left, but the third character is relevant because player 2 leaves at that moment and advances one unit. The fourth character means that player one advances one more unit completing the distance D , so the time for player 1 is 4. The fifth character is 2, so the player 2 also finishes the race and his time is 3 (he finished when the total time was 5 but he left at moment 3, in other words, he was in race during characters 3, 4 and 5 — don't forget that this race is against the clock!). The last character could be ignored.

The program should output the classification of the participants in accordance to the time registered when they get to the finish line, but there's one restriction: the simulation is valid only if all players have finished the race and travel times have been normally distributed with an index of asymmetry lesser than 20%, which is defined as follows:

$$A_s = \left| 3 \cdot \frac{(\mu - median)}{\sigma} \right|$$

Input

The input consists of several test cases. The first line of each one contains an integer T , the number of simulations ($1 \leq T \leq 1000$). Each simulation starts with 2 integers N ($2 \leq N \leq 6$) and D ($1 \leq D \leq 10$). The next line contains the string S with length L , representing the stopwatch. It is granted that the string only contains valid numbers, that is, $1 \leq S_i \leq N$.

After that, there is a line that contains Q ($1 \leq Q \leq \frac{L}{4} + 1$). The following line will contain Q different values from X ($1 \leq X \leq \frac{L}{4} + 1$), each one separated by an space.

Output

If the arrival times of cyclists have been normally distributed, print the order of arrival, otherwise display a message indicating that the simulation was invalid. Check the examples to see how to show that. If two or more players have the same time, order by the player's number.

Sample Input

```
2
4 1
21243
1
1
2 2
122121111111
3
1 2 3
```

Sample Output

```
ORDER OF RACING: 4 1 2 3
ORDER OF RACING: 2 1
ORDER OF RACING: 2 1
INVALID SIMULATION
```