

# Schedules

PEPE the singer has released a new song. He wants to control how many times his songs is played on RUMBA FM radio station. PEPE has hired two companies (A and B) to perform this task. Every day, they give him a schedule indicating the time when the song has been played on the radio. Song always takes the same number of seconds to play, so end times are not important. We are only interested in start times.

On the first day PEPE received both schedules. They were almost identical. He verified that each entry of A corresponds to exactly one entry in B. He simply took a pencil, and he marked one entry in A, and then the corresponding entry in B. He folled with this approach as long as unmarked entries exist. The second day PEPE again received both schedules, but he found that the number of entries in both schedules is not the same. Moreover, the times did not match at all due to a human error. He said, "Oh Gosh!, I have to conciliate both schedules, finding the best possible match between them". He only trusts the entries that can be matched in both schedules. But how to match them? PEPE started by deciding how many seconds of error (difference) he is able to tolerate for two matched entries. Then he tried to find the largest number of possible matches. For equally large machings, he is interested in smallest average time difference in seconds. Unfortunately, it may take too long since his new song is vere popular, having many hits in RUMBA FM. So, we need your help to perform this task automatically.

## Input

The input consists in several test cases. Each test case starts with a line containing three integer numbers  $N_a$ ,  $N_b$ , and  $S$ , separated by single spaces.  $N_a$  and  $N_b$  are the number of entries in A and B respectively ( $1 \leq N_a, N_b \leq 200$ ), and  $S$  is the tolerance in seconds ( $0 \leq S \leq 7200$ ). The second line contains  $N_a$  time stamps in the format hh:mm:ss separated by single spaces. The third line contains  $N_b$  time stamps in same format as the previous one. Note that all start times are in the same day.

The end of the input is an empty test case, where  $N_a = N_b = S = 0$  and should not be processed.

## Output

For each test case, the output is a single line containing an integer  $K$  and floating point number  $V$  rounder to one decimal place.  $K$  is the largest number of matches between schedule A and schedule B.  $V$  is the average time difference in seconds between the  $K$  matched entries. In case there is no possible match, your program should instead print "No matches"

## Example

### Input:

```
4 2 120
03:00:00 03:00:59 07:40:00 12:40:04
02:59:14 12:41:45
3 2 60
03:00:59 07:40:00 12:40:04
```

02:59:14 12:41:45  
0 0 0

**Output:**  
2 73.5  
No matches