In order to ensure that the contestants find it easy to reach the regional contest site, the authority has prepared some robot-driven vehicles. The vehicles would visit n predetermined junctions and carry the contestants waiting there to the contest. There is a computer controlled Transportation Center (TC) that determines the number of seats of each vehicle and the time each vehicle leaves the contest site for the first time.

When a new vehicle is needed, a request is made to the TC. Every new vehicle has fewer number of seats than the last vehicle if it is more than 3: the i-th vehicle has $\max(s-(i-1)*t, 3)$ seats $(i = 1, 2, \ldots)$. The first vehicle leaves the contest site just at 8:00am. When TC gets a request for a new vehicle, it prepares a new vehicle and right after 2 seconds of getting the request, the new vehicle leaves the contest site. If multiple requests are made at the same time, only one is considered.

At the junction j, each vehicle does the following tasks. If there are more than one vehicle at j at the same time, they perform the tasks in order of their service times: the one with the longest service time goes first. Service time of a vehicle is the difference between current time and the time the vehicle left the contest site (which is at junction 0) for the first time.

1. If $j = 0$ (the contest site), all the contestants in the vehicle gets down. Otherwise, the vehicle picks up as many contestants as it can (i.e. until the vehicle is full or there are no contestants left at junction j).
2. If, after that, there are any contestant left at junction j $(j > 0)$, the vehicle sends a request for a new vehicle to the TC.
3. Finally the vehicle starts moving towards the next junction k, which is selected by the robot-driver in the following way (even at junction 0):
   - If the vehicle is full, $k = 0$.
   - Otherwise, if no other vehicle has left junction j yet, $k = (j + 1) \bmod n$.
   - Otherwise, k is $((k_0 + 1) \bmod n)$ if it is different from j.
   - Otherwise, k is $((k_0 + 2) \bmod n)$.
     Here, k_0 is the '*next junction*' selected by the last vehicle leaving junction j.

Vehicles do the above 3 tasks instantly (i.e. in 0 seconds). Time needed to go from each junction to any other junction is known. All the contestants reach a suitable junction by 8:00am and don't go away until they are picked up by any vehicle. Given the number of contestants waiting at each junction and a time limit, you are to determine when everyone reaches the contest or how many have reached the contest by the time limit.

Input

Input consists of several datasets. Each dataset consists of the followings:

- A line containing the name of the set (which has 2 to 20 alphanumeric characters).
- A line containing 3 positive integers n, s and t $(2 < n < 11)$.
- Each of next n lines contains $n-1$ integers each. The i-th line $(i = 1, 2, 3 \ldots)$ contains the time (in seconds) needed to go from the junction $i-1$ to all other junctions (except the $(i-1)^{th}$) in order $0, 1, 2 \ldots n$.
- Next $n-1$ lines each contain a non-negative integer. The i-th line $(i = 1, 2, 3 \ldots)$ contains the number of contestants waiting at the i-th junction.
- The last line of the dataset contains the time limit (in seconds, less than 10000000).

All integers on a single line are separated by exactly one space. Total number of contestants is at most 1000.

The end of input is marked with a line consisting of '`TheEnd`'.

Output

For each set, print 2 lines. The first line contains the name of the set as it appears in the input. The second line contains the time (in seconds) needed to bring all the contestants to the contest, if the time is not more than the given time limit. Otherwise, print the number of contestants reached the contest by the time limit.

Sample Input

```
Dhaka2000
3 22 4
30 8
10 30
28 8
20
20
100
Dhaka2001
3 22 4
30 8
10 30
28 8
20
20
90
Dhaka2002
3 22 2
30 8
10 30
28 8
20
20
100
TheEnd
```

Sample Output

```
Dhaka2000
98 seconds needed
Dhaka2001
22 contestants reached
Dhaka2002
88 seconds needed
```