ACM Intercollegiate Programming Contest
Pacific NW Region
1999

Problem B

Gone Fishing

John is going on a fishing trip. He has *h* hours available (1 <= h <= 16), and there are *n* lakes in the area (2 <= n <= 25), *L_1, ..., L_n*, all reachable along a single, one−way road. John starts at lake 1, but he can finish at any lake he wants. He can only travel from one lake to the next one, but he does not have to stop at any lake unless he wishes to. For each *i = 1, ..., n−1*, the number of 5−minute intervals it takes to travel from *L_i* to *L_(i+1)* is denoted *t_i* (0 < *t_i* <= 192). For example, *t_3 = 4* means that it takes 20 minutes to travel from *L_3* to *L_4*

To help plan his fishing trip, he has gathered some information about the lakes. For each lake *L_i*, the number of fish expected to be caught in the initial 5 minutes, denoted *F_i* (*F_i* >= 0), is known. Each 5 minutes of fishing decreases the number of fish expected to be caught in the next 5−minute interval by a constant rate of *d_i* (*d_i* >= 0). If the number of fish expected to be caught in an interval is less than *d_i*, there will be no more fish left in the lake in the next interval. To simplify the planning, John assumes that no one else will be fishing at the lakes to affect the number of fish in the lakes.

Write a program to help John plan his fishing trip to maximize the number of fish expected to be caught. The number of minutes spent fishing at each lake must be a multiple of 5.

Input

You will be given a number of cases in the input. Each case starts with a line containing *n*. This is followed by a line containing *h*. This is followed by a line of *n* integers specifying *F_i* (*i = 1, ..., n*), followed by a line of *n* integers *d_i* (*i = 1, ..., n*), and finally, a line of *n−1* integers *t_i* (*i = 1, ..., n−1*). Input is terminated by a case in which *n* = 0.

Output

For each test case, print the number of minutes spent fishing at each lake, separated by commas, for the plan achieving the maximum number of fish expected to be caught (you should print the entire plan on one line even if it exceeds 80 characters). This is followed by a line containing the number of fish expected. If multiple plans exist, choose the one that spends as long as possible at *L_1*, even if no fish are expected to be caught in some intervals. If there is still a tie, choose the one that spends as long as possible at *L_2*, and so on. Insert a blank line between cases.

Sample Input

```
2
1
10 1
2 5
2
4
4
10 15 20 17
0 3 4 3
1 2 3
4
4
10 15 50 30
0 3 4 3
1 2 3
0
```

Sample Output

```
45, 5
Number of fish expected: 31

240, 0, 0, 0
Number of fish expected: 480

115, 10, 50, 35
Number of fish expected: 724
```