Problem G

Fast Food

Source: `fastfood.(c|cc|pas|java)`
Input: `fastfood.in`

The fastfood chain McBurger owns several restaurants along a highway. Recently, they have decided to build several depots along the highway, each one located at a restaurent and supplying several of the restaurants with the needed ingredients. Naturally, these depots should be placed so that the average distance between a restaurant and its assigned depot is minimized. You are to write a program that computes the optimal positions and assignments of the depots.

To make this more precise, the management of McBurger has issued the following specification: You will be given the positions of n restaurants along the highway as n integers $d_1 < d_2 < ... < d_n$ (these are the distances measured from the company's headquarter, which happens to be at the same highway). Furthermore, a number k ($k \leq n$) will be given, the number of depots to be built.

The k depots will be built at the locations of k different restaurants. Each restaurant will be assigned to the closest depot, from which it will then receive its supplies. To minimize shipping costs, the *total distance sum*, defined as

$$\sum_{i=1}^{n} \mid d_i - (\text{position of depot serving restaurant } i) \mid$$

must be as small as possible.

Write a program that computes the positions of the k depots, such that the total distance sum is minimized.

Input

The input file contains several descriptions of fastfood chains. Each description starts with a line containing the two integers n and k. n and k will satisfy $1 \leq n \leq 200, 1 \leq k \leq 30, k \leq n$. Following this will n lines containing one integer each, giving the positions d_i of the restaurants, ordered increasingly.

The input file will end with a case starting with $n = k = 0$. This case should not be processed.

Output

For each chain, first output the number of the chain. Then output an optimal placement of the depots as follows: for each depot output a line containing its position and the range of restaurants it serves. If there is more than one optimal solution, output any of them. After the depot descriptions output a line containing the total distance sum, as defined in the problem text.

Output a blank line after each test case.

Sample Input

```
6 3
5
6
12
19
20
27
0 0
```

Sample Output

```
Chain 1
Depot 1 at restaurant 2 serves restaurants 1 to 3
Depot 2 at restaurant 4 serves restaurants 4 to 5
Depot 3 at restaurant 6 serves restaurant 6
Total distance sum = 8
```