ACM International Collegiate Programming Contest 2006

South American Regional Contests

November 10-11, 2006

Warmup Session

This problem set contains 2 problems; pages are numbered from 1 to 4.

This problem set is used in simultaneous contests hosted in the following countries:

- Argentina
- Bolivia
- Brazil
- Chile
- Colombia
- Peru
- Venezuela

Problem A
Soccer League

Source file name: `soccer.c,` `soccer.cpp`, `soccer.java` *or* `soccer.pas`

Soccer, or football, is today the biggest sport in the world. According to a study published in 2001, over 240 million people regularly play football in more than 200 countries in every part of the world.

You have been hired to provide IT support for the ACM Soccer League. The rules set for this year in the ACM Soccer League are:

- The winner Club of a League Match shall score three points. Each Club participating in a League Match which is drawn shall score one point.

- The results of League Matches shall be recorded in a Table containing, in respect of each Club, the number of points scored in that Season.

- The position of Clubs in the table shall be determined by the number of points scored in that Season, the Club having scored the highest number of points being at the top of the table and the Club having scored the lowest number of points being at the bottom.

- If any two or more Clubs have scored the same number of points their position in the table shall be determined on goal difference, that is to say, the difference between the total number of goals scored by and against a Club in League Matches in that Season, and the higher or highest placed Club shall be the Club with the larger or largest goal difference.

You will be given a list of match results from the ACM Soccer League, and must produce League Table, as defined by the rules above.

Input

The input contains several test cases. The first line of a test case contains a single integer N indicating the number of matches in the test ($1 \le N \le 1000$). Each of the following N lines describes the result of a match, in the format

```
Club1 X x Y Club2
```

where `Club1` and `Club2` are strings of at most 15 characters (from 'a' to 'z' and 'A' to 'Z'), `X` is the number of goals scored by `Club1` and `Y` is the number of goals scored by `Club2` ($0 \le$ `X` ≤ 10 and $0 \le$ `Y` ≤ 10). The end of input is indicated by $N = 0$.

The input must be read from standard input.

Output

For each test case in the input your program should output a League Table. Each line of the table must be in the format

```
P Club
```

where `P` is the number of points accumulated by the `Club`. The Clubs must appear in order of descending number of points. If two or more Clubs have the same number of points, the one with largest goal difference must appear first. If two or more teams have the same number of points and the same goal difference, you can choose to output them in any order. You must print a blank line after each table.

The output must be written to standard output.

| Sample input	Example of output for the sample input
3 Warriors 0 x 3 Sundance Sundance 3 x 2 Cats Cats 2 x 2 Warriors 6 Goalies 1 x 1 Wallops Wallops 1 x 1 Guardian Guardian 1 x 1 Goalies Artists 1 x 1 Goalies Artists 1 x 1 Guardian Wallops 1 x 1 Artists 0	6 Sundance 1 Cats 1 Warriors 3 Goalies 3 Wallops 3 Guardian 3 Artists

Problem B
Tornado!

Source file name: `tornado.c,  tornado.cpp`, `tornado.java` *or* `tornado.pas`

Is this crazy weather the result of mankind's continuous interference in the environment? Or is it simply the normal cycle of climate changes through the ages? No one seems to know for sure, but the fact is that natural phenomena such as tornadoes and hurricanes have been hitting our country with more force and frequence than in past decades.

One tornado has just hit Silverado Farm, a cattle and milk producer, and made havoc. The barn roof was torn, several trees were uprooted, the farm truck was overturned... But the worst thing is that the tornado destroyed several sections of the fence that surrounded the property. The fence was very well built, with concrete posts every two meters, and barbed wire enclosing the whole farm perimeter (the perimeter, in meters, is an even number, making the fence perfectly regular).

Now several posts are broken or missing, and there are gaps in the fence. To prevent the cattle from getting out of the property, the fence must be restored as quickly as possible. Reconstructing the fence to its original form, with concrete posts, will take a long time. In the meantime, the farm owners decided to close the gaps with a temporary fence, made with wooden posts. Wooden posts will be placed in exactly the same spots where missing/broken concrete posts were/are. However, in order to make the temporary reconstruction faster and less expensive, the owners decided to use fewer posts: a wooden post will be used to replace a missing/broken concrete post only if the length of the barbed wired needed to close the distance to the next post (wooden or concrete) exceeds four meters.
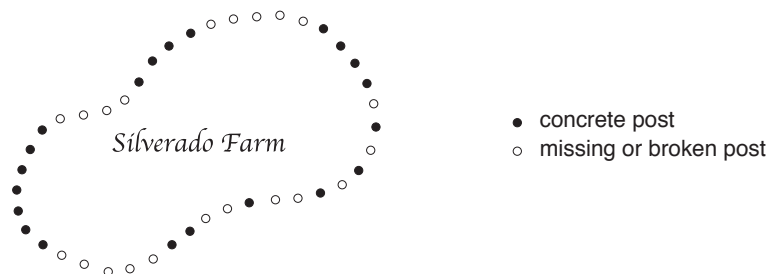
Given the description of which posts are missing/broken, you must write a program to determine the smallest number of wooden posts needed to close all the gaps in the fence, according to the owners' decision.

Input

The input contains several test cases. The first line of a test case contains one integer N indicating the number of original concrete posts in the fence ($5 \le N \le 5000$). The second line of a test case contains N numbers X_i indicating the state of each concrete post after the tornado ($0 \le X_i \le 1$ for $1 \le i \le N$). If $X_i = 1$ post i is in good condition, if $X_i = 0$ post i is broken or missing. Note that post X_N is next to post X_1. The end of input is indicated by $N = 0$.

The input must be read from standard input.

Output

For each test case in the input your program must produce one line of output, containing an integer indicating the smallest number of wooden posts that are needed to restore the fence, according to the owners' decision.

The output must be written to standard output.

Sample input	Output for the sample input
10 1 0 0 1 0 0 1 0 1 1 11 1 0 0 1 0 0 0 1 1 0 1 12 0 0 0 0 0 1 1 0 0 0 1 1 0	2 2 3