National Computer Programming Contest 2000

7th October, 2000
Bangladesh University of Engineering & Technology

Sponsored by
Ministry of Science & Technology

We more frequently fail to face the right problem
than fail to solve the problem we face.

Rules:

1. There are eight (8) problems for each team to be completed in five hours.
2. All problems require you to read test data from the input file specified in the problem and write results to the standard output.
3. Output must correspond exactly to the provided sample output format, including (mis)spelling and spacing. Multiple spaces will not be used in any of the judges' output, except where explicitly stated.
4. Programming style is not considered in this contest. You are free to code in whatever style you prefer. Documentation is not required.
5. Contestant teams will submit their program through the PC^2 software.
6. Students can use their books, papers, documentation, source codes of programs. But no soft copy will be allowed.
7. Judges' decisions are to be considered final.

Tips:

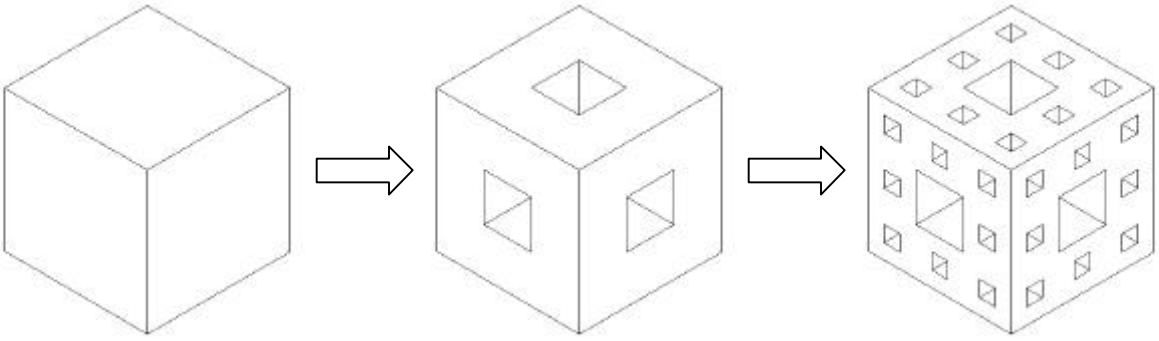
Try to solve the easiest problem first.

Problem A

Simulating a Sponge

Input : sponge.in
Output : standard output

Sponge is manufactured by adding foaming agents to melted plastic or melted rubber. A small amount of plastic or rubber produces a large amount of foam. There is an approximate mathematical model to represent a sponge. The model starts with a cubical block of side *l* as shown in the first figure below. Then a square hole of size *l*/3 is drilled across each face – three holes in total, as shown in second figure. Afterwards, another hole of size *l*/9 is drilled along the middle of each sub-cube generated after the first drill, as shown in the third figure. The next drill, not shown in the figure, will be of size *l*/27 across each sub-sub-cube generated after the second operation. It is said, that if operation is continued to infinity, the final entity will have a volume of zero, but a surface area of infinity.



In this problem, starting from a cube of side *l*, your job is to find the area and volume of the resulting entity after *n* levels of drilling operation as described above.

Input

The input is a series of lines each containing two nonnegative integers. The first integer is *l* and the second one is *n*. The input terminates with a line containing two zeros for *l* and *n*, which must not be processed.

Output

For each line in the input print the surface area and volume of the resulting entity after the given number of drilling levels. Both figures must be converted to the nearest smaller integer. The input data is such that the final surface area is less than 10^{14}. Output for each test case must be on a separate line.

Sample Input

```
3 3
2 0
9 5
0 0
```

Sample Output

```
200 10
24 8
7536 162
```

Problem B

Tom Catches Jerry

Input : tom.in
Output : standard output

Jerry is sleeping in the bed of Tom at floor b of a multi-storied building. When Tom is at floor a, he discovers Jerry in his bed through the video display. After a cheerful smile, he decides to catch Jerry in his flat. The building has n floors and m elevators. Each elevator has a predefined schedule $S_i = \{f_p\ f_q\ \ldots\ f_t\}$ where i is the elevator number (identification number), $f_p\ f_q\ \ldots\ f_t$ are the floors of the building and $p < q < \ldots < t$. In the schedule, p and t are the boundary floors of the elevator. Each elevator can move upward or downward touching each floor listed in its schedule. In each intermediate stoppage, it holds for 30 sec. to load and/or unload. When it reaches any of the boundary floors in its schedule (i.e., f_p or f_q in the above schedule), it holds there for one minute and then begins its next trip in the reverse direction. The time required to move from the k-th floor to the $k + 1$-th (or, $k - 1$-th) floor is 10 sec. and it is uniform (i.e., for an l-floor move it requires $10 \times l$ seconds).

At the time of watching Jerry in his bed, Tom found that the elevators have just reached the following floors: $e_1, e_2, e_3, \ldots, e_{i-1}, e_i, e_{i+1}, \ldots, e_m$, where e_i is the floor reached by the i-th elevator. A positive e_i indicates that the elevator reached the floor from downwards, and a negative e_i indicates that it reached that floor from upwards.

The problem is to find the minimum time (in seconds) required by Tom to catch Jerry in floor b. Assume that 0 (zero) time is required to switch from one elevator to another.

Input

The input may contain several test cases. The first line will contain an integer indicating the number of test cases to follow.

The first line of each test case will contain four (4) integers: n (number of floors), m (number of elevators), a (Tom's floor number) and b (Jerry's floor number), separated by one or more spaces. You may assume that $0 < n \le 256$, $0 < m \le 64$, $1 \le a, b \le n$.

The second line will contain m integers separated by one or more spaces. The i-th ($1 \le i \le m$) of these integers will indicate the position and direction of the i-th elevator when Tom decided to catch Jerry.

The i-th ($1 \le i \le m$) of the next m lines will contain the floor schedule of elevator i. The first integer in the line is the number of floors in the schedule followed by the floor numbers in increasing order.

Two consecutive test cases will be separated by one blank line.

Output

The output will contain one line for each test case. Each line will contain a number indicating the minimum time (in seconds) required by Tom to catch Jerry. If Tom has no possibility of catching Jerry then output a -1.

Sample Input

```
2
20 5 7 3
-1 6 2 -8 1
```

```
7 1 4 7 10 13 16 19
5 3 5 6 10 20
6 1 2 5 9 11 14
3 3 8 18
8 1 5 7 8 10 12 14 17

10 3 1 10
-1 2 -8
3 1 5 9
4 1 2 3 4
3 3 5 8
```

Sample Output

```
520
-1
```

Problem C

Take the Trees

Input : trees.in
Output : standard output

A poor man went to the King. Citing his pecuniary condition the man asked for some wealth so that he can survive with his wife and children. The King said, “I give you some plants to grow. If you take care they will give you shelter and food. There are plants in the nursery nearby. In each shade there is a known number of plants. Initially, all the plants are unmarked. You should arrange all of them in a rectangular grid such that no grid point in that rectangle remains empty and count the number of plants along its length and breadth. Now mark the same number of plants in the grid which were marked at most once before. For example, 20 plants can be arranged in a 4×5 rectangular grid that has 5 plants along its length and 4 along its breadth. So select $4 + 5 = 9$ plants from the grid which were never marked more than once before and mark them. Now arrange the plants in a rectangle of different size along the grids, and mark plants according to the rule stated above. For example, the 20 plants in the previous example can also be arranged in a 2×10 rectangular grid. Continue marking plants in this way until no further arrangement is possible. Only if you can mark twice each of the plants of the same shade, you will be given all of them.”

There are several shades in the nursery with known number of plants. Help the poor man in identifying the shade from which he can pick all the plants.

Input

The first line of the input contains an integer *n* giving the number of shades in the nursery. Each of the next *n* lines will contain the number of plants in a shade.

Output

For each shade in the input print “Yes” if the poor man can pick up trees from this shade, “No” otherwise. Each output must be on a separate line.

Sample Input

```
3
6
28
17
```

Sample Output

```
Yes
Yes
No
```

Problem D

Binary Stars and Black Holes

Input : stars.in
Output : standard output

Black Holes are astronomical bodies that are remnants of some dead stars. After the star has burned all its gases, the stars collapse. The force of collapse in some cases is so great that electrons of the atoms strip from the orbits, and collapse on to the nucleus of the atom. The resulting matter is extremely dense. It is said that if our earth were to collapse to such an extent the radius of the earth would be 1 centimeter only. These bodies are called *Black Holes*. The gravitational force of a Black Hole is so great that nothing, even light, can escape from its surface. The reason that even light does not escape from a *Black Hole* poses an interesting problem.

Black Holes cannot be seen. Therefore, they are detected by observing the orbits of nearby stars. Stars must be in orbit around another heavier body. Therefore, if a star is observed for which no other body is observed in the center of its orbit, it is conjectured that there is a black hole in the center. Even though, this is straight forward, it does also pose another possibility that the star may be part of a *Binary Star System* in which two stars rotate around each other with a common center. Therefore, to detect a *Back Hole*, orbit measurements of two stars are conducted. Then one of the following possibilities are encountered:

a) if the center of rotations of the two stars are different, the stars are independent star systems;

b) if the center of rotations of the two stars are same, then
   i) if the stars are rotating independently, there is a *Black Hole* at the center;
   ii) if the stars and the center are always on one straight line, the two stars form a *Binary Star System*.

In this problem, for the position of the stars appearing in the data set, you are to determine if there is a *Black Hole* somewhere, or if the star system is a *Binary Star*, or the star system is rotating independently. You are to assume that all trajectories are circular. Furthermore, consider two floating-point values equal if the absolute value of their difference is less than 0.1.

Input

The input may contain multiple test cases. Each test case consists of six lines. Each line contains two floating-point numbers. The first three lines of data consists of *x* and *y* coordinates of the first star measured at three different times. The next three lines contain the *x* and *y* coordinates of the second star measured at the same times.

Output

For each test case in the input output a line stating whether there is a *Black Hole* or the star system is a *Binary Star* or rotating independently. Strictly follow the format shown in the sample output. Separate two consecutive words by a single space. All floating-point numbers must be printed correct to one place after the decimal point.

Sample Input

```
63.7963 12.5591
19.8578 75.3097
```

```
-32.7963 12.5592
25.3481 23.7635
18.0882 15.8407
12.9118 15.8408
70.7404 45.1824
12.8177 85.7404
-19.4577 7.82131
11.6519 34.7635
23.2364 26.6519
30.1603 41.5
64.7404 41.1824
6.81766 81.7404
-25.4577 3.82131
10.1519 13.2635
21.7364 5.15192
28.6603 20
```

Sample Output

```
There is a Black Hole at 15.5, 26.5.
There is a Binary Star system with center at 21.5, 36.5.
Independent star system.
```

Problem E

Register Allocation

Input : register.in
Output : standard output

A computer program stores the values of its variables in memory. For arithmetic comparisons, the values must be entered in registers. Registers are expensive, so we want to use them efficiently. If two variables are not used at the same time we can allocate them to the same register. For each variable, we compute the first and last time when it is used. A variable is active during the interval between these two times.

From past experience, a system analyst knows that he always needs exactly five registers to store the values of all variables. Unfortunately, he does not know how to divide the variables into five disjoint groups so that no two variables of the same group are active at the same time. You are requested to help him in solving the problem.

Input

The input may contain multiple test cases. The first line of each test case contains an integer n (≤ 500). Each of the next n lines contains the name of a variable followed by the first and last time (measured from the start of the execution of the program) when it is used.

Variable names will be at most 2 characters long and will not contain any white-space character. The variable name, the first time and the last time are separated by spaces.

Input terminates with a value 0 for n.

Output

For each test case in the input first output the test case number (starting from 1) as shown in the sample output. Then for each i (= 1, 2, 3, 4, 5) print an integer m_i on line i indicating the number of variables to be assigned to register i, and the names of those variables with every two consecutive variable names separated by a single space. Note that the solution is not unique, and hence any valid solution is acceptable.

Print a blank line between the output of two consecutive test cases.

Sample Input

```
6
A 1 6
B 2 8
C 3 7
D 2 6
E 5 10
F 7 10
8
A 1 5
B 4 9
C 2 10
D 7 15
E 6 11
F 8 13
```

```
G 12 16
H 14 17
0
```

Sample Output

```
Case #1
2 A F
1 B
1 C
1 D
1 E

Case #2
3 A E H
1 F
2 B G
1 C
1 D
```

Problem F

Who is the King?

Input : king.in
Output : standard output

There is an isolated island in the *Bay of Bengal*. The inhabitants are too irritated by their brutal king and they wanted to replace him. One day, at late night some of them attacked the palace and killed the king. In the morning, the Prime Minister announced the death of the king. Unfortunately the king had no successor. So, the Prime Minister wanted to select a king from the inhabitants. But none of them agreed to be the king of the island. So, the Prime Minister decided to wait until any one comes to the island. They decided the first person who will arrive on the island would be the king of that island.

Fortunately, on next day new faces arrived on the island but they were three in number. So the Prime Minister requested them to choose one of them as the king. After a long discussion, three friends reached a solution that each of them will be the king in turn. But the turn will be selected in a specific manner.

The king will be changed on every lunar eclipse. But if there is a solar eclipse on the same day then the king will remain in his position. Suppose the lunar eclipse occurs every 150 days and the solar eclipse occurs on every 300 days. You will be given the arrival date of the three friends and you have to find out who is the king at a given date. Last date of solar and lunar eclipse was 11 Oct, 1999.

Input

The first line of input contains an integer giving the number of test cases to consider. Each test case consists of a single line containing the arrival date of the three persons *A*, *B* and *C*, the query date and the sequence of the three persons for the selection of kingship. (for example, a sequence *A C B* means that A will be the first king, then *C*, then *B*, then *A* again and so on)

The dates will be in the following format:

day-of-month month-name, year

Here, denotes a single space. Month name will be one of the followings: *Jan*, *Feb*, *Mar*, *Apr*, *May*, *Jun*, *Jul*, *Aug*, *Sep*, *Oct*, *Nov* and *Dec*. You may assume that all dates will be valid and there will be no date earlier than 11 Oct, 1999. No query will be made for a date before the arrival of the three friends.

Output

For each test case in the input, output the name of the king on the query date. Each output must be on a separate line.

Sample Input

```
1
18 Feb, 2000 15 Nov, 2000 A C B
```

Sample Output

```
c
```

Problem G

Test the Rods Economically

Input : rods.in
Output : standard output

National Construction and Project Centre (*NCPC*) and the *Bureau of Civil Engineering Works* (*BCEW*) have been given the authority of testing and certifying the quality of rods used in construction works in the country. The *Get and Do* construction company has recently got a contract of construction at different sites of the country. Before the construction can start they want to get the rods from their n sites tested either at *NCPC* or at *BCEW*. *Get and Do* has got the permission of testing T_1 rods at *NCPC* and T_2 at *BCEW*. There are m_i samples at site i ($1 \le i \le n$). Sum total of these samples over all the n sites is just equal to ($T_1 + T_2$). The cost of testing j items from site i at *NCPC* is $C_{i,j,1}$ and that of testing at *BCEW* is $C_{i,j,2}$. Write a program to find a minimum cost testing schedule for the *Get and Do* company.

Input

The input may contain multiple test cases. The first line of each test case contains the two nonnegative integers T_1 and T_2 ($1 \le T_1 + T_2 \le 300$). The next line contains n ($1 \le n \le 30$). Then follow $3n$ lines. For $1 \le i \le n$, line ($3i$ - 2) contains the value of m_i ($1 \le m_i \le 20$), line ($3i$ - 1) contains m_i nonnegative integers $C_{i,j,1}$ ($1 \le j \le m_i$) and line $3i$ contains m_i nonnegative integers $C_{i,j,2}$ ($1 \le j \le m_i$). You may assume that $0 \le C_{i,j,1}, C_{i,j,2} \le 1000$.

A test case containing two zeros for T_1 and T_2 terminates the input, and this case must not be processed.

Output

For each test case in the input print two lines. The first line contains an integer giving the minimum cost for testing all the samples at *NCPC* and *BCEW*. The next line contains n integers with two consecutive integers separated by a single space. The i-th integer gives the number of samples from site i that are tested at *NCPC* (it is implicit that the rest are tested at *BCEW*). Note that the second output line is not unique, and hence any optimal testing schedule is acceptable. Print a blank line between the output of two consecutive test cases.

Sample Input

```
10 12
5
5
10 30 70 150 310
10 20 40 60 180
7
30 60 90 120 160 200 240
20 60 100 130 160 200 240
4
40 60 80 100
30 70 100 120
3
60 120 180
20 50 90
3
30 70 100
```

```
30 70 100
0 0
```

Sample Output

```
580
1 3 4 0 2
```

Problem H

Get the Multiplicity of Factors

Input : factor.in
Output : standard output

Given a positive integer *n* you need to write a program that counts the number of times an integer *m* appears as a factor in the factorial of *n*.

Input

There will be several lines of input each containing two positive integers *n* and *m* ($0 < m < n < 2^{32}$). The input terminates with two zeros for *n* and *m*.

Output

For each line in the input print a line containing an integer representing the multiplicity of *m* in *n*.

Sample Input

```
6 2
8 3
8 2
0 0
```

Sample Output

```
4
2
7
```