ACM Asia Regional (Kanpur Site) Programming Contest 2001

Rules:

1. There are eight (8) problems for each team to be completed in five (5) hours.

2. All problems require you to read test data from the standard input specified in the problem and write results to the standard output.

3. You can use any of the standard library functions that your chosen program programming language provides. In addition, you can use the math library in C/C++. You can not use any other library that requires an extra flag to be passed to the compiler command. If you do this, the judges will probably find a code "compilation error" in your program.

4. Your program is not permitted to invoke any external programs. For example, you cannot use in C the call system ("grep xyz abc") to determine whether the file abc contains an occurrence of the string xyz. *Violation of this rule may lead to disqualification from the contest.*

5. The name of the file containing the sample input given for each problem. If you test your program using PC^2, it will automatically redirect input from the sample input file to your program.

6. 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.

7. Your solution to any problem should be submitted for judging using the PC^2 software only. Once you have submitted the solution, it will reach the judges. The time it takes for your problem to be judged will depend on how busy the judges are. Once your submission has been judged, you will receive a message through PC^2 indicating the judgement. The judgement maybe "Yes", meaning that your submission was judged to be correct. Otherwise you will get a message indicating the problem with your program. For example, the message may be "Incorrect Output", "Output Format Error", "Compilation Error", "Runtime Error", "Run Time Limit Exceeded" etc.

8. Programming style is not considered in this contest. You are free to code in whatever style you prefer. Documentation is not required. The judges will only test whether the input−output behavior of your program is correct or not.
   However, if your program takes more than 3 minutes to execute for some input, it will be assumed to have gone into an infinite loop and judged incorrect.

9. Contestants can use their books, papers, documentation, source codes of programs. But no soft copy will be allowed. No team is allowed to co−operate with any other team, using any means whatsoever, for solving the contest problems. *Any such attempt, if detected, will lead to immediate disqualification of all the teams involved.*

10. Judges' decisions are to be considered final.

Note: *Extra Blank−spaces, Tabs, Blank−lines, Commas, Quotes will be ignored from the standard output. For output format, you are advised to see carefully sample output.*

Problem A

A Round Table Conference

Input : dislikes.in
Output : standard output

Leaders of N political parties agree to have a round table conference to resolve a political issue through discussion. However due to animosity between parties and leaders some / all leaders may dislike sitting next to leaders specified by them. The problem is to determine, if possible, sitting arrangements around the round table so that a leader is not required to sit next to another leader disliked by him / her.

Leaders are identified by integers 1,2,.....N. It may be noted that if a leader I dislikes sitting next to another leader J then the leader J may not necessarily dislike sitting next to I. However in such a case I and J can not sit next to each other because I dislikes J. Further, clock−wise and anti clock−wise arrangements are considered identical.

An arrangement is represented by a sequence of N integers 1,2,..... N indicating the relative positions of each leader beginning with leader 1. It is assumed that the last integer in the sequence and the integer 1 appearing in the first position are next to each other in the sitting arrangement. The arrangements are to be printed in lexicographic order.

Write a program to print all possible sitting arrangements for the round table conference.

Input

The input may contain multiple test cases.

The first line of each test case contains the case number c and the total number of leaders N (<100).

Each of the next N lines contains N zeros / ones.

If d_{ij} denotes the j^{th} entry in the i^{th} line then d_{ij} is equal to 1 if leader i dislikes leader j, otherwise d_{ij} is equal to 0. It is assumed that d_{ii} is equal to 1. This is true for all i,,j=1,2,..... N.

The input terminates with an input 0 for c.

Output

For each test case in the input, first print, in a line, the test case number c and the total number k ($k \geq 0$) of possible arrangements as shown in sample output.

Each of the next k lines prints an arrangement in lexicographic order.

Print a blank line between two successive test cases.

Sample Input

```
1 5
1 1 0 1 0
0 1 0 0 1
0 0 1 0 0
1 0 0 1 0
0 1 0 1 1
2 6
1 0 1 1 0 0
0 1 0 0 0 1
1 0 1 0 0 0
1 0 0 1 1 0
0 0 1 0 1 1
0 1 0 0 1 1
0
```

Sample Output

```
1 0

2 2
1 5 2 3 4 6
1 5 2 4 3 6
```

Problem B

A Reputed Doctor's Prescription

Input : petmedicines.in
Output : standard output

A doctor has a great reputation for his treatments. However his prescriptions are illegible. He alone and only his own chemist can read his prescriptions. Sometimes it is difficult even to identify the same medicine written in two different prescriptions.

The doctor has a standard pattern of prescribing medicines. He has a standard set of N pet medicines outside which he never prescribes any medicine. For every pet medicine there is a set of related successor pet medicines. A pet medicine may appear in one or more sets of successor pet medicines.

The doctor prescribes medicines in the form of a list. If he prescribes a medicine then he never repeats it in the same prescription. Further, when he prescribes a medicine then the next medicine, if any, in the prescription is one of the successor pet medicines and none of these successor pet medicines appears in the prescription subsequently.

A team of handwriting experts and medical practitioners analyzed a large number of prescriptions of the doctor and made a possible list of his pet medicines. The medicines in this list are identified by N integers 1,2,?N. For each pet medicine in the list, they also prepared a list of successor pet medicines one of which possibly succeeds the medicine in every prescription.

A prescription has a list of p medicines. If some of the medicines in a prescription are identified then it is required to identify the unidentified medicines in the prescription.

Write a program that will identify the unidentified medicines if one or more medicines in the list are given as identified.

Input

The input may contain multiple test cases with data from one or more expert teams.

For each test case, the first line contains the expert team number n and the number N (<500), of pet medicines identified by the team.

Each of the next N lines contains a set of integers: the i^{th} line identifies the successor pet medicines of the i^{th} pet medicine.

Each test case is related to a given prescription. There is a line that gives the number p. Next p lines identify some of the p medicines in the prescription. The i^{th} line contains j if the i^{th} medicine in the prescription is identified as j. If the i^{th} medicine is unidentified then j is equal to 0.

The entire input set terminates with an input 0 for n.

Output

For each test case in the input, print a line containing the expert team number n and the total number k ($k \geq 0$) of possible solutions. Each of the next k lines contains a complete sequence of p possible medicines in lexicographic order.

Print a blank line between two successive test cases.

Sample Input

```
1 5
2 4 5
1 3
5
2 3
1 2 4
4
0
3
0
0
2 6
3 4
1 5 6
2 4 6
1 3
2 6
2 4 5
4
4
0
2
1
0
```

Sample Output

```
1 2
2 3 5 4
4 3 5 1

2 0
```

<u>Problem C</u>

Marriages in DAMPU community

Input : matchmaking.in
Output : standard output

A community named DAMPU lives in the twenty first century. However it still follows some of its old outdated traditional community customs very strictly, even though its members are used to modern way of life and use modern gadgets including computers for its activities and recreations.

The DAMPU community is closed in the sense that marriages take place according to the old custom between a male and a female member of the community. A member of the community is not permitted to marry outside the community. Further, members of the community are monogamous. Each unmarried living member having a specified marriageable age is eligible for marriage. An eligible member of the community can marry another eligible member of opposite sex if and only if, at least one of them is separated from the nearest common ancestor by at least k generations, k being an integer specified by the community council from time to time. The integer k is called the *generation gap*. For example if k=2 then the nearest common ancestor should be either grand parents of at least one of the two members who wish to marry or ancestors of grand parents. It may be noted that the members who are already married may or may not have a *generation gap* of k generations because a different criterion was possibly applicable at the time when they were married.

The DAMPU community wants to develop a computer program for computer based match making so that a member eligible for marriage may easily get a list of eligible members of the opposite sex with whom there is a possibility of marriage. It is known that the DAMPU community maintains a register for births, deaths and marriages and the records are available for every member of the community.

Can you help the DAMPU community by writing a program for the purpose stated above?

Input

The input may contain multiple test cases.

For each test case, the first line contains three integers n, N and k. The integer n is the case number, while N (< 5000) is the total number of members of the community for whom data are given and k ($k \leq 100$) is the specification fixed by the community council for the *generation gap* to be used for match making.

Each of the next N lines contains relevant data (viz., Sex, Father, Mother) on members of the community selected for testing the program. The i^{th} line gives data for the i^{th} member. The members of the community for whom data are given are identified by integers 1,2,..,N. These members include all living members who are eligible for marriage and also some ancestors of eligible members so that the eligibility for match making can be verified. Integers 1and 0 are used to denote the sex of a member: 1, if male and 0, if female. A parent of a member is denoted by the integer 0 if the generation gap between the parent and every eligible member, other than his/her descendents, is known to be k or more.

The entire input set terminates with an input 0 0 0 for n, N and k.

Output

For each test case in the input, first print in one line, the test case number n and the total number m of possible matches. Print a line for each possible match in the form: i j where i is a male member and j is a female member eligible for marriage. The list of matches should be printed in lexicographic order of i and j.

Keep a blank line before printing the output for the next test case, as shown in the sample output.

Sample Input

```
1 9 2
1 0 0
1 0 0
0 2 0
1 1 5
0 2 0
1 1 5
0 2 0
1 6 3
0 1 5
2 18 3
1 0 0
1 0 0
1 1 6
0 0 0
0 3 4
0 0 0
0 2 14
1 12 13
0 15 7
1 15 7
1 3 4
1 2 14
0 3 4
0 0 0
1 12 13
0 12 13
1 15 7
0 15 7
0 0 0
```

Sample Output

```
1 3
4 7
8 7
8 9

2 4
10 5
11 9
11 18
```

17 5

Problem D

Road Network in a New Township

Input : roadsignals.in
Output : standard output

A new township has been built over a rectangular plot of land dividing the whole area by a grid. The grid divides the whole plot into (m−1) x (n−1) big square plots for use as residential / commercial / industrial areas. Each square plot is divided into smaller plots and these are well connected by internal road network. However, the major road links in the township connecting all big square plots are the avenues running along the grid with m x n junctions. The junctions are assumed to be arranged in m rows and n columns and are denoted by (p, q), p=1, 2, ?m , q=1, 2, ?.n. For an internal junction (p, q), the neighboring junction (p−1, q) is assumed to be at the north of (p, q). In addition to the road network of the township there is a rectangular perimeter road that runs around the township with connections from each road junction located at the boundary of the township. An outline of the road links in the township is shown below where the perimeter road is indicated by the rectangle drawn around the network and a road junction at (p,q) with directions to four roads going out of the junction are indicated by

```
     ↑
←(p, q)→ .
     ↓
```

```
                 <−−  Perimeter road  −−>
                          North
   ↑          ↑              ↑                  ↑
←(1,1)→  ←(1,2)→       ←(1,−)→          ←(1,n)→
   ↓          ↓              ↓                  ↓

   ↑          ↑              ↑                  ↑
←(2,1)→  ←(2,2)→       ←(2,−)→          ←(2,n)→
   ↓          ↓              ↓                  ↓
??????????????????..??????..
   ↑          ↑              ↑                  ↑
←(−,1)→  ←(−,2)→       ←(−,−)→          ←(−,n)→
   ↓          ↓              ↓                  ↓
   ↑          ↑              ↑                  ↑
←(m,1)→  ←(m,2)→       ←(m,−)→          ←(m,n)→
   ↓          ↓              ↓                  ↓
```

The traffic control department of the township enforces static traffic control rules that either permit or do not permit traffic out of a road junction in a given direction. At each road junction there are four traffic signals, one facing each of the four roads connected at the junction. Each of these signals is lighted either Green or Red according to whether traffic is or is not permitted *out of the junction*. A traffic in the perimeter road is permitted to enter the township through a road junction connecting the perimeter road if the traffic signal at the junction facing the perimeter road, is lighted Green.

It is generally believed that moving along the perimeter road is more risky than moving along the township avenues. It is also believed that the risk while moving from a source junction to a neighboring destination junction in the township avenues is same whatever be the location of the junctions. Let this risk be R. The risk is additive in the sense that if one moves from a source junction to a destination junction through the township avenues, passing through k−1 other junctions then the risk is equal to kR. When the perimeter road is used to move from a junction to another junction separated by k−1 other junctions ion the boundary of the township then the risk increases by a constant factor f and the risk is equal to fkR, where f is a real constant that depends on the time of journey. In addition, for moving along the corners of the perimeter road there is an additional risk, it is estimated to be 2fR for each corner. Let the total risk in moving along a rout be denoted by FR. The factor F is called the risk factor of the rout.

As an illustration it may be pointed out that the risk for moving from (1,2) to (2,1) is 2R if the movement is through the rout (1,2), (1,1), (2,1). So the risk factor F of this rout is equal to 2. However, the risk becomes 4fR (=2fR+2fR) if one moves along the rout (1,2), perimeter, (2,1) crossing the corner of the perimeter road once. In this case if f is defined to be 2.5 then the risk factor F is equal to 10.0.

Given a source junction (a,b) and a destination junction (c,d) when certain traffic control rules are under operation and the constant f is known, the problem is to find the total number s of the safest routs to move from the source to the destination and determine the risk factor F associated with the safest routs.

Write a program to solve this problem.

Input

The input may contain multiple test cases.

For each test case, the first line contains the case number c, the total number of rows m, and columns n, the total number r of {source, destination} pair for each of which the safest routs are to be found and the constant factor f.

Each of the next r lines gives a, b, c, d indicating the {source (a,b) , destination (c,d)} pair for which the safest routs are to be found.

The traffic signals at m x n junctions are given in row major order of the junctions in the next m x n lines. The traffic signals at a junction are represented by a string of four 0’s and 1’s, 0 representing a Green signal and 1 representing a Red signal. The signals are given in clock−wise order starting from the signal at the north. Thus if 0011 represents the four signals at a junction then the signal at the north is Green, the signal at the east is also Green while two other signals are Red.

The entire input set terminates with an input 0 for the case number c.

Output

For each test case print the test case number c in one line. For each given {source (a,b), destination (c,d)} pair, print a, b, c, d, s and F as shown in the sample output. Print a blank line before printing output for the next test case.

Sample Input

```
1 4 5 2 1.5
2 2 1 1
1 1 4 1
0101
1001
0011
0011
1100
1010
1010
1001
1100
0110
0110
0011
1100
0011
0110
1010
0110
0110
0101
0101
2 4 6 1 2
2 1 3 5
0011
0011
1101
1011
1001
1101
0011
0111
0100
0110
1011
0101
1100
0110
1110
0110
0110
1000
1011
0001
0011
0101
0110
1110
0
```

Sample Output

```
1
2 2 1 1 1 5.5
```

```
1 1 4 1 1 4.0

2
2 1 3 5 6 19.0
```

Problem E

Viewer's Prize in F−TV

Input : viewer.in

Output : standard output

The successful launching of third−generation (3G) mobile phone by the Japanese giant DoCoMo marked the beginning of an era in real−time multi−media inter−net communications at affordable cost. The F−TV or the Future TV has the reputation to venture into innovative TV programs that might attract large number of viewers in near future. In order to remain ahead in the race to have the largest number of viewers during a prime time slot in comparison to other TV channels, the management of F−TV plans to produce a TV program that will provide a large number of viewers with an opportunity to chat with a celebrity either in person or in real−time virtually using mobile phones and get cash prizes in the process.

The planned TV program consists in having a celebrity on each show as the anchor of the chat show and a selected number of viewers as 'real' participants. Any number of remote viewers may also join the program during the show as 'virtual' participants using their 3G mobile phones from anywhere in the world. Each participant, male or female, real or virtual, is enrolled for the show and is given a unique token serial number on first come first served basis. Towards the end of the show the anchor runs a computer program that selects participants from among the enrolled participants and announces cash prizes for the selected participants. There is a possibility that no participant is selected for a prize; in such a case the anchor announces the decision accordingly.

The computer program selects the participants to get prizes in a simple way. Let n be the total number of participants, real or virtual. Each participant is given a token serial number p, p=1, 2, ?, n. Let $x_{p,}$ be equal to 0 if the participant having the token serial number p, is female, otherwise x_p is equal to 1. The input to the program is the string x=(x_1, x_2, ? x_n) of length n on the alphabet { 0, 1}. The program finds the longest palindromes contained in x (i.e. the longest sub−strings made up of elements of x each of which reads the same backward as forward). Let there be k such palindromes and let U be the union of all these palindromes. The program deletes U from x. It repeats the process with the reduced string x successively until there is no palindrome of length 3 or more. Finally it finds the elements of x, if any, which are retained in the process.

For example, let n=37 and x=(0110 1110 1101 1011 0111 0101 1101 0010 0101 1) where the elements of x are written in groups of four followed by a blank character, for easy readability. The sub−string (x_3,?x_{22})=(10 1110 1101 1011 0111 01) is of length 20 and is the longest palindromes in x. If this sub−string is deleted then the string x is reduced to (x_1, x_2, x_{23}, x_{24},?. x_{37}) with elements (0101 1101 0010 0101 1) which when reduced further gives (x_1, x_2, x_{23},x_{24}) with elements (0101). Finally, since the longest palindromes in (0101) are (010) and (101) , each of which is of length 3 and their union is the entire string under consideration, all the four elements are deleted. Hence, no element of x is selected.

Input:

The input may contain multiple test cases.

For each test case, the first line contains two integers the case number c and the length n of the given string.

The string x is given in one or more lines in groups of four elements separated by a blank character; the last group in the string may contain less than four elements. The string is terminated with the character $ which is not considered as a part of the string.

The entire input set terminates with an input 0 for c.

Output:

For each test case in the input, print the test case number c and the number k of participants selected for the prize. In next k lines, print the token serial numbers of the participants selected for the prize in ascending order.

Print a blank line before printing the output of the next test case.

Sample Input:

```
1  37
0110  1110 1101  1011  0111  0101  1101  0010  0101 1 $
2  62
1011 0001 1110 1010 1111 0010 1010 0111 1101 1010 0110 0011 0110 0011 0111 10 $
0
```

Sample Output:

```
1 0

2 1
1
```

Problem F

Execution of a Will

Input : valuation.in
Output : standard output

Mr. Paisawalla died at the age of 79. He was a rich man. He left behind lots of immovable property. Before his death, he made a will for disposition of his property to his wife, three sons and a daughter. The directions in the will were as follows:

1. The property should be disposed of impartially and equally to his wife, three sons and the daughter according to current valuation of property and practical limitations.
2. As far as practicable equality in shares should be maintained. However, if complete equality is not feasible, then the largest difference between shares of any two members should be the least. In such a case the disposal should be in decreasing order of shares to wife, son1, son2, son3 and the daughter.
3. The impartiality in valuation is to be ensured by an impartial committee for valuation.
4. The equality and impartiality in disposition are to be ensured by entrusting the job to a software professional who should write a program for taking computer−based decisions.

As an illustration let the property of late Mr. Paisawalla be as indicated in the table below:

Property No	1	2	3	4	5	6	7	8	9
Valuation in lakhs	49	78	83	35	10	109	95	23	62

The disposition should be as follows:

Members:	Wife	Son1	Son2	Son3	Daughter
Property No	2, 4	1, 9	6	3, 8	5, 7
Share	113	111	109	106	105

Are you prepared to write a computer program for disposition of the property of late Mr. Paisawalla? The program should be able to provide solutions to other similar cases.

Assume that the total number of receivers to whom the property is to be distributed is m and the receivers are represented by integers 1, 2,?m. In case equality in disposition is not feasible the disposition should be such that share of property decreases as the number representing a receiver increases.

Input:

The input may contain multiple test cases for testing the program.

For each test case, the first line contains three integers, the case number c, the total number n of property and the total number m of receivers to whom the property is to be distributed.

For simplicity, properties are identified by integers 1, 2, ??n. The valuations of property are given in n lines, the i^{th} line gives the valuation of the i^{th} property.

The entire input set terminates with an input 0 for c.

Output:

For each test case in the input, print the test case number c and the difference between the largest and the smallest allocation of shares.

Write shares of the property allocated to m receivers in the next m lines. The i^th line contains the total valuation of shares received by the i^th receiver, together with the list of property in increasing order of the number representing the property, as shown in the sample output.

Sample Input:

```
1 9 5
49
78
83
35
10
109
95
23
62
2 11 4
83
142
75
93
43
5
27
18
129
62
12
0
```

Sample Output:

```
1 8
113 2 4
111 1 9
109 6
106 3 8
105 5 7

2  1
173  3  4  6
172  5  9
172  1  7 10
172  2  8 11
```

Problem G

Editing a Book

Input : passages.in
Output : standard output

Modern writers do not use pens to write books. They enter the text directly on to the home PCs using MS Word and edit the text when necessary.

After entering the text for a new book, a famous novelist desires to have a major rearrangement of the text he has prepared. He has identified passages in the text and numbered these passages. However the passages are not in proper sequence. He wants to use the minimum number of **cut** and **paste** operations of MS Word to edit the text and put the passages in proper order. He may do the operation either with one passage or with a number of passages occurring in a sequence. He needs a program for this purpose.

The following example illustrates his problem. In order to edit the text containing 6 passages in the sequence P2, P4, P1, P5, P3, P6 so that the passages appear in the text in the proper sequence viz., P1, P2, P3, P4, P5, P6 he needs at least 2 **cut** and **paste** operations:

1. **cut** P1 and **paste** before P2 to get P1, P2, P4, P5, P3, P6

and 2. **cut** P3 and **paste** after P2 to get P1, P2, P3, P4, P5, P6.

Can you write a program for him?

Input:

The input may contain multiple test cases.

For each test case, the first line contains two integers, the case number c and the total number n of passages.

For simplicity, passages are represented by integers 1, 2, ??n. The input sequence of passages is given in n lines, each line containing one integer in the order in which the passages appear in the text before editing.

The entire input set terminates with an input 0 for c.

Output:

For each test case in the input, print the test case number c and the total number k of **cut** and **paste** operations needed for the rearrangement of passages.

Print the given sequence in one line. In the next k lines print successive changes in the sequence of passages due to application of k **cut** and **paste** operations.

Print a blank line between outputs of two test cases.

Sample Input:

```
1 6
2
4
1
5
3
6
2 15
12
6
7
2
3
4
9
10
11
1
5
8
13
14
15
0
```

Sample Output:

```
1 2
2 4 1 5 3 6
1 2 4 5 3 6
1 2 3 4 5 6

2 4
12  6   7    2    3    4   9  10  11    1     5    8   13   14   15
12  6   7    9  10  11   1    2   3     4     5    8   13   14   15
12  9 10  11    1   2   3    4    5     6     7    8   13   14   15
12  1   2   3    4    5   6    7    8     9   10  11   13   14   15
1    2   3   4    5    6   7    8    9   10   11  12   13   14   15
```

<u>Problem H</u>

The Most Wanted Man in the History of Mankind

Input : spot.in
Output : standard output

There is a search around the world for the most wanted man (MWM) in the history of mankind. He is believed to have an extraordinary quality, either superior or inferior, compared to a normal person. He is likely to be present any where in the world. If he is spotted precisely then he may be captured without much difficulty. However the problem is in spotting him because a number of look−alikes are known to exist at different parts of the world.

Apart from conventional resources, modern satellite communication and surveillance systems that can precisely identify, measure, record and monitor the physical, social, cultural and behavioral characteristics of a person, are available for collecting data on the MWM. In addition, the services of competent professionals in different disciplines are available to analyze and interpret data collected from all sources.

Professional Psychologists and Statisticians admit that it is difficult for them to distinguish precisely between two persons using physical and behavioral data. However, by analyzing and interpreting group data, they claim that they can differentiate precisely between a group of normal persons and a similar group of same size that includes a person of extraordinary quality even if it is **not known** in advance whether the extraordinary quality is superior or inferior to that of a normal person. In such a case for a reliable comparison it is necessary that the number of persons in each group is 3 or more.

In other words, let A, B be two groups of equal number of persons, the number being 3 or more. An unidentified person x having an extraordinary quality classified either superior or inferior, may or may not be present in any of the groups. It is possible to perform a **special compare operation on A, B** and state precisely by data analysis and interpretation, one of the following:

(a) A=B indicating x is neither in A nor in B,
(b) A>B indicating either x is in A and he is superior or x is in B and he is inferior,
(c) A<B indicating either x is in A and he is inferior or x is in B and he is superior.

The outcome of the special compare operation may be denoted by E, H and L representing the cases (a), (b) and (c) respectively.

Since collection, analysis and interpretation of data are considered costly and the total number n of look−alikes including the MWM is small, $8 \le n \le 12$, it is decided that only three special compare operations on selected groups could be performed successively to spot the MWM and determine his qualitative characteristic. However it is required to select groups A_i, B_i, i= 1,2,3, before each special compare operation**.** The groups may depend on the outcome of the previous special compare operations.

Write a program that selects the groups to be compared by the professionals in three stages. For each stage, you may assume that the outcomes of special compare operations are known for all previous stages. After selection of groups the program should spot the MWM when successive outcomes of special compare operations are known.

Input:

The input may contain multiple test cases.

The data for a test case are given in a single line. The line contains two integers, the case number c and the total number of look−alikes, $8 \leq n \leq 12$. It also contains a string S of three letters s_1, s_2, s_3 representing the successive outcomes of three special compare operations to be used by your program to form the groups for comparison and to spot the MWM.

The look−alikes, including the MWM w, are identified by integers 1, 2, ??n and each of the outcomes s_1, s_2, s_3 of the special compare operations is represented by the letters E, H or L. The i^{th} letter s_i in the string S represents the outcome of the i^{th} special compare operation, i=1,2,3.

The entire input set terminates with an input 0 for c.

Output:

For each test case in the input, print four lines.

The first line contains the input data, viz., c, n, S, together with an integer w that spots the MWM and a letter q that identifies his quality. If MWM is spotted then $1 \leq w \leq n$ and the letter q is either s or i depending on whether the quality of MWM is identified as superior or inferior. If MWM is not among the look−alikes then w is equal to zero and the letter q is equal to m, representing 'MWM missing'. In case the comparisons reveal that the claims of the professionals are faulty, then w is equal to −1 and q is equal to f representing 'faulty comparisons and claims are not tenable'.

The next three lines contain A_i, s_i, B_i, i =1, 2, 3 where A_i, B_i are the groups for comparison at the i^{th} stage and s_i is the given outcome of the special compare operation.

Print a blank line between outputs of two test cases.

Sample Input:

```
1 8 HHH
2 8 HHE
3 8 HHL
4 8 HEH
5 8 HEE
6 8 HEL
7 8 HLH
8 8 EEE
0
```

Sample Output:

```
1 8 HHH 1 s
1 2 3 H 4 5 6
1 4 7 H 2 5 8
1 5 6 H 2 7 8

2 8 HHE −1 f
1 2 3 H 4 5 6
1 4 7 H 2 5 8
1 5 6 E 2 7 8



3 8 HHL  5 i
1 2 3 H 4 5 6
```

```
1 4 7 H 2 5 8
1 5 6 L 2 7 8

4 8 HEH −1 f
1 2 3 H 4 5 6
1 4 7 E 2 5 8
1 5 6 H 2 7 8

5 8 HEE  3 s
1 2 3 H 4 5 6
1 4 7 E 2 5 8
1 5 6 E 2 7 8

6 8 HEL  6 i
1 2 3 H 4 5 6
1 4 7 E 2 5 8
1 5 6 L 2 7 8

7 8 HLH −1 f
1 2 3 H 4 5 6
1 4 7 L 2 5 8
1 5 6 H 2 7 8

8 8 EEE  0 m
1 2 3 E 4 5 6
1 4 7 E 2 5 8
1 5 6 E 2 7 8
```