Nationwide Computer Programming Contest

Electro Computer Days 2000

15th June, 2000

Sponsored by
IBM World Trade Corporation

God grant me the serenity

To accept the problems that I cannot solve

The persistence to solve the problems that I can

And the wisdom to know the difference.

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.

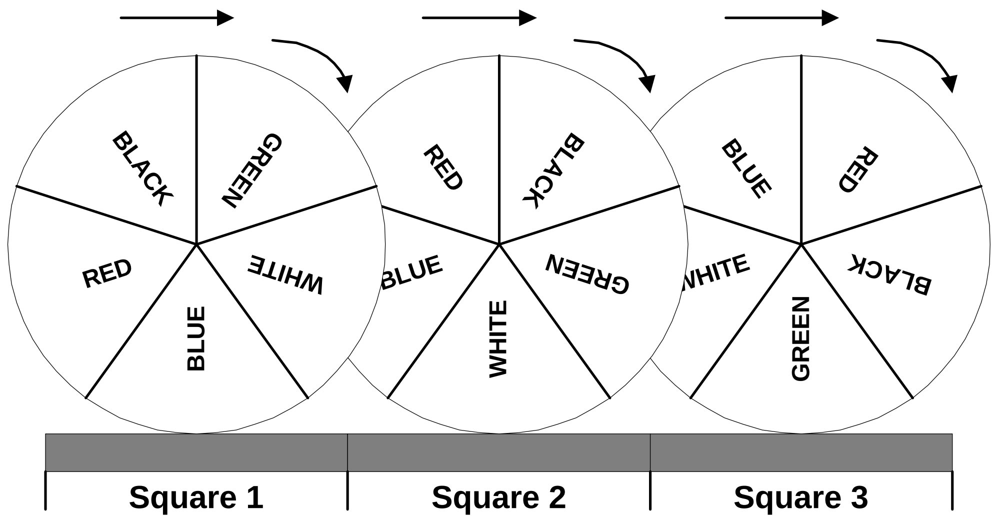
Tips:

Try to solve the easiest problem first.

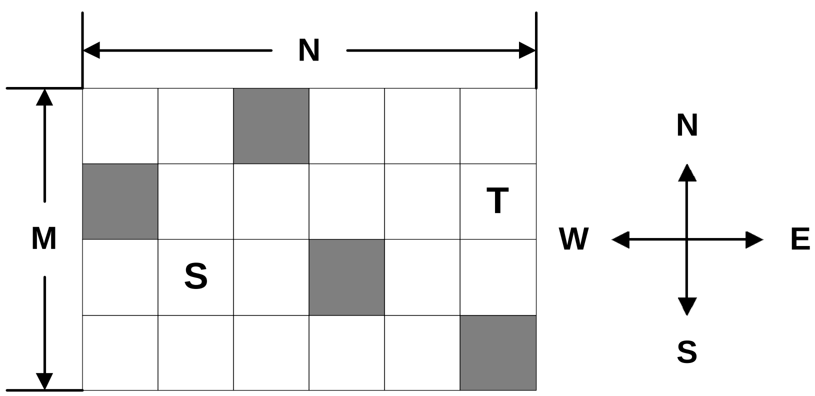
Problem A
The Monocycle

Input : cycle.in
Output : standard output

A monocycle is a cycle that runs on one wheel and the one we will be considering is a bit more special. It has a solid wheel colored with five different colors as shown in the figure:



The colored segments make equal angles (72°) at the center. A monocyclist rides this cycle on an $\boldsymbol{M} \times \boldsymbol{N}$ grid of square tiles. The tiles have such size that moving forward from the center of one tile to that of the next one makes the wheel rotate exactly 72° around its own center. The effect is shown in the above figure. When the wheel is at the center of square 1, the mid-point of the periphery of its blue segment is in touch with the ground. But when the wheel moves forward to the center of the next square (square 2) the mid-point of its white segment touches the ground.



Some of the squares of the grid are blocked and hence the cyclist cannot move to them. The cyclist starts from some square and tries to move to a target square in minimum amount of time. From any square either he moves forward to the next square or he remains in the same square but turns 90° left or right. Each of these actions requires exactly 1 second to execute. He always starts his ride facing north and with the mid-point of the green segment of his wheel touching the ground. In the target square, too, the green segment must be touching the ground but he does not care about the direction he will be facing.

Before he starts his ride, please help him find out whether the destination is reachable and if so the minimum amount of time he will require to reach it.

Input

The input may contain multiple test cases.

The first line of each test case contains two integers ***M*** and ***N*** ($1 \leq M, N \leq 25$) giving the dimensions of the grid. Then follows the description of the grid in ***M*** lines of ***N*** characters each. The character '#' will indicate a blocked square, all other squares are free. The starting location of the cyclist is marked by 'S' and the target is marked by 'T'.

The input terminates with two zeros for ***M*** and ***N***.

Output

For each test case in the input first print the test case number on a separate line as shown in the sample output. If the target location can be reached by the cyclist print the minimum amount of time (in seconds) required to reach it exactly in the format shown in the sample output, otherwise, print "destination not reachable".

Print a blank line between two successive test cases.

Sample Input

```
1 3
S#T
10 10
#S.......#
#..#.##.##
#.##.##.##
.#....##.#
##.##..#.#
#..#.##...
#......##.
..##.##...
#.###...#.
#.....###T
0 0
```

Sample Output

```
Case #1
destination not reachable

Case #2
minimum time = 49 sec
```

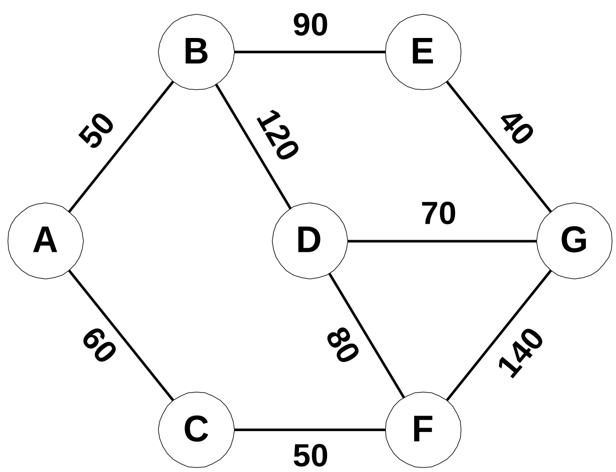
Problem B
Audiophobia

Input : audio.in
Output : standard output

Consider yourself lucky! Consider yourself lucky to be still breathing and having fun participating in this contest. But we apprehend that many of your descendants may not have this luxury. For, as you know, we are the dwellers of one of the most polluted cities on earth. Pollution is everywhere, both in the environment and in society and our lack of consciousness is simply aggravating the situation.

However, for the time being, we will consider only one type of pollution **-** the sound pollution. The loudness or intensity level of sound is usually measured in *decibels* and sound having intensity level 130 decibels or higher is considered painful. The intensity level of normal conversation is 60-65 decibels and that of heavy traffic is 70-80 decibels.

Consider the following city map where the edges refer to streets and the nodes refer to crossings. The integer on each edge is the average intensity level of sound (in decibels) in the corresponding street.



To get from crossing **A** to crossing **G** you may follow the following path: **A-C-F-G**. In that case you must be capable of tolerating sound intensity as high as 140 decibels. For the paths **A-B-E-G**, **A-B-D-G** and **A-C-F-D-G** you must tolerate respectively 90, 120 and 80 decibels of sound intensity. There are other paths, too. However, it is clear that **A-C-F-D-G** is the most comfortable path since it does not demand you to tolerate more than 80 decibels.

In this problem, given a city map you are required to determine the minimum sound intensity level you must be able to tolerate in order to get from a given crossing to another.

Input

The input may contain multiple test cases.

The first line of each test case contains three integers C (≤ 100), S (≤ 1000) and Q (≤ 10000) where C indicates the number of crossings (crossings are numbered using distinct integers ranging from 1 to C), S represents the number of streets and Q is the number of queries.

Each of the next S lines contains three integers: c_1, c_2 and d indicating that the average sound intensity level on the street connecting the crossings c_1 and c_2 ($c_1 \neq c_2$) is d decibels.

Each of the next Q lines contains two integers c_1 and c_2 ($c_1 \neq c_2$) asking for the minimum sound intensity level you must be able to tolerate in order to get from crossing c_1 to crossing c_2.

The input will terminate with three zeros form C, S and Q.

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 query in the input print a line giving the minimum sound intensity level (in decibels) you must be able to tolerate in order to get from the first to the second crossing in the query. If there exists no path between them just print the line "no path ".

Print a blank line between two consecutive test cases.

Sample Input

```
7 9 3
1 2 50
1 3 60
2 4 120
2 5 90
3 6 50
4 6 80
4 7 70
5 7 40
6 7 140
1 7
2 6
6 2
7 6 3
1 2 50
1 3 60
2 4 120
3 6 50
4 6 80
5 7 40
7 5
1 7
2 4
0 0 0
```

Sample Output

```
Case #1
80
60
60

Case #2
40
no path
80
```

Problem C
Self-describing Sequence

Input : sequence.in
Output : standard output

Solomon Golomb's *self-describing sequence* $\langle f(1), f(2), f(3), \ldots\rangle$ is the only non-decreasing sequence of positive integers with the property that it contains exactly $f(k)$ occurrences of k for each k. A few moments thought reveals that the sequence must begin as follows:

n	1	2	3	4	5	6	7	8	9	10	11	12
$f(n)$	1	2	2	3	3	4	4	4	5	5	5	6

In this problem you are expected to write a program that calculates the value of $f(n)$ given the value of n.

Input

The input may contain multiple test cases. Each test case occupies a separate line and contains an integer n ($1 \le n \le 2{,}000{,}000{,}000$). The input terminates with a test case containing a value 0 for n and this case must not be processed.

Output

For each test case in the input output the value of $f(n)$ on a separate line.

Sample Input

```
100
9999
123456
1000000000
0
```

Sample Output

```
21
356
1479
1684
438744
```

Problem D
Hartals

Input : hartals.in
Output : standard output

A social research organization has determined a simple set of parameters to simulate the behavior of the political parties of our country. One of the parameters is a positive integer ***h*** (called the *hartal parameter*) that denotes the average number of days between two successive *hartals* (strikes) called by the corresponding party. Though the parameter is far too simple to be flawless, it can still be used to forecast the damages caused by *hartals*. The following example will give you a clear idea:

Consider three political parties. Assume $h_1 = 3$, $h_2 = 4$ and $h_3 = 8$ where h_i is the *hartal parameter* for party i ($i = 1, 2, 3$). Now, we will simulate the behavior of these three parties for $N = 14$ days. One must always start the simulation on a Sunday and assume that there will be no *hartals* on weekly holidays (on Fridays and Saturdays).

Days	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
	Su	**Mo**	**Tu**	**We**	**Th**	**Fr**	**Sa**	**Su**	**Mo**	**Tu**	**We**	**Th**	**Fr**	**Sa**
Party 1			×			×			×			×		
Party 2				×				×				×		
Party 3								×						
Hartals			**1**	**2**				**3**	**4**			**5**		

The simulation above shows that there will be exactly 5 *hartals* (on days 3, 4, 8, 9 and 12) in 14 days. There will be no *hartal* on day 6 since it is a Friday. Hence we lose 5 working days in 2 weeks.

In this problem, given the *hartal parameters* for several political parties and the value of ***N***, your job is to determine the number of working days we lose in those ***N*** days.

Input

The first line of the input consists of a single integer ***T*** giving the number of test cases to follow.

The first line of each test case contains an integer ***N*** ($7 \le N \le 3650$) giving the number of days over which the simulation must be run. The next line contains another integer ***P*** ($1 \le P \le 100$) representing the number of political parties in this case. The ***i***-th of the next ***P*** lines contains a positive integer h_i (which will never be a multiple of 7) giving the *hartal parameter* for party ***i*** ($1 \le i \le P$).

Output

For each test case in the input output the number of working days we lose. Each output must be on a separate line.

Sample Input

```
2
14
3
```

```
3
4
8
100
4
12
15
25
40
```

Sample Output

```
5
15
```

Problem E
Tower of Cubes

Input : cubes.in
Output : standard output

In this problem you are given ***N*** colorful cubes each having a distinct weight. Each face of a cube is colored with one color. Your job is to build a tower using the cubes you have subject to the following restrictions:

- Never put a heavier cube on a lighter one.
- The bottom face of every cube (except the bottom cube, which is lying on the floor) must have the same color as the top face of the cube below it.
- Construct the tallest tower possible.

Input

The input may contain multiple test cases. The first line of each test case contains an integer N ($1 \le N \le 500$) indicating the number of cubes you are given. The i-th ($1 \le i \le N$) of the next N lines contains the description of the i-th cube. A cube is described by giving the colors of its faces in the following order: front, back, right, left, bottom, and top face. For your convenience colors are identified by integers in the range 1 to 100. You may assume that cubes are given in the increasing order of their weights, that is, cube 1 is the lightest and cube N is the heaviest.

The input terminates with a value 0 for N.

Output

For each test case in the input first print the test case number on a separate line as shown in the sample output. On the next line print the number of cubes in the tallest tower you have built. From the next line describe the cubes in your tower from top to bottom with one description per line. Each description contains an integer (giving the serial number of this cube in the input) followed by a single white-space character and then the identification string (front, back, left, right, top or bottom) of the top face of the cube in the tower. Note that there may be multiple solutions and any one of them is acceptable.

Print a blank line between two successive test cases.

Sample Input

```
3
1 2 2 2 1 2
3 3 3 3 3 3
3 2 1 1 1 1
10
1 5 10 3 6 5
2 6 7 3 6 9
5 7 3 2 1 9
1 3 3 5 8 10
6 6 2 2 4 4
1 2 3 4 5 6
10 9 8 7 6 5
```

```
6 1 2 3 4 7
1 2 3 3 2 1
3 2 1 1 2 3
0
```

Sample Output

```
Case #1
2
2 front
3 front

Case #2
8
1 bottom
2 back
3 right
4 left
6 top
8 front
9 front
10 top
```

Problem F
Inviting Politicians

Input : invite.in
Output : standard output

Before the next parliamentary election the president is going to have a meeting with the top politicians of the country. But, as you know, our politicians are usually arrogant in nature and can't tolerate opposition. So, if all them are invited to the same meeting, they will ruin it (the same way they are ruining the country). However, from his past experience the president knows that he can always divide the politicians into four disjoint groups such that no two fighting politicians are placed in the same group. Then each of the four groups will be invited on a different day and thus avoiding any unpleasant situation.

The president requests you to help him in solving the problem.

Input

The first line of the input contains an integer T (≤ 15) indicating the number of test cases to follow.

The first line of each test case contains two integers N (≤ 300) and M (≤ 5000). Each of the next N lines contains the name of a politician. No name will be more than 10 characters long and will not contain any white-space character. Each of the next M lines contains the name of two politicians (both of which are valid names occurring in the names-list given at the beginning of the test case) who are not in good terms with each other and hence can not be placed in the same group.

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) print an integer P_i on line $2i$ indicating the number of politicians to be invited on day i, and on line $2i + 1$ print the names of those politicians with every two consecutive names separated by a single white-space character. Assume that the politicians can always be invited in 4 days and can never be invited in less than 4 days. Note that the solution is not unique and hence any valid solution is acceptable.

Print a blank line between two consecutive test cases.

Sample Input

```
2
4 6
A
B
C
D
A B
A C
A D
B C
B D
C D
6 7
A
```

```
B
C
D
E
F
A B
A C
A D
B C
B D
C D
D E
```

Sample Output

```
Case #1
1
A
1
B
1
C
1
D

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

Problem G
Envelopes

Input : envelope.in
Output : standard output

I have bought some greeting cards for my friends but in order to send them I must also buy some envelopes. Each card must be put inside a separate envelope without bending or tearing. The envelopes are made of so thin papers that one can put inside an envelope a card having even the same dimensions as that envelope.

Please help me choose envelopes so that the total amount I need to spend in buying them is minimized.

Input

The input may contain multiple test cases.

The first line of each test case contains two integers M ($1 \le M \le 5$) and N ($M \le N \le 10$) where M is the number of greeting cards and N is the number of envelopes to choose from. The i-th ($1 \le i \le M$) of the next M lines consists of two integers l_i and w_i ($1 \le l_i, w_i \le 50000$) giving the dimensions of the i-th greeting card. The i-th ($1 \le i \le N$) of the next N lines contains three integers L_i, W_i and C_i ($1 \le L_i, W_i, C_i \le 50000$) where L_i and W_i give the dimensions of the i-th envelope and C_i is its price in Taka.

The input terminates with two zeros for M and N.

Output

For each test case in the input first print the test case number on a separate line as shown in the sample output. If an envelope can be chosen for each of the greeting cards in the input, print one line for each where the i-th line contains the number of the envelope (in the order given in the input) inside which the i-th greeting card should be put. Otherwise, print "cannot buy" on a line by itself. Note that if there are multiple solutions with minimum cost, any one of them is acceptable.

Print a blank line between two successive test cases.

Sample Input

```
2 4
105 9
99 149
110 10 10
100 50 5
150 100 7
90 140 15
1 2
100 150
99 149 10
152 100 5
0 0
```

Sample Output

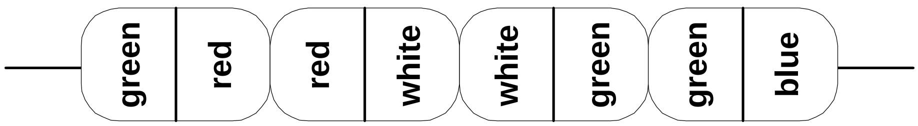
```
Case #1
2
3

Case #2
cannot buy
```

Problem H
The Necklace

Input : necklace.in
Output : standard output

My little sister had a beautiful necklace made of colorful beads. Two successive beads in the necklace shared a common color at their meeting point. The figure below shows a segment of the necklace:



But, alas! One day, the necklace was torn and the beads were all scattered over the floor. My sister did her best to recollect all the beads from the floor, but she is not sure whether she was able to collect all of them. Now, she has come to me for help. She wants to know whether it is possible to make a necklace using all the beads she has in the same way her original necklace was made and if so in which order the bids must be put.

Please help me write a program to solve the problem.

Input

The input contains T test cases. The first line of the input contains the integer T.

The first line of each test case contains an integer N ($5 \le N \le 1000$) giving the number of beads my sister was able to collect. Each of the next N lines contains two integers describing the colors of a bead. Colors are represented by integers ranging from 1 to 50.

Output

For each test case in the input first output the test case number as shown in the sample output. Then if you apprehend that some beads may be lost just print the sentence "some beads may be lost" on a line by itself. Otherwise, print N lines with a single bead description on each line. Each bead description consists of two integers giving the colors of its two ends. For $1 \le i \le N - 1$, the second integer on line i must be the same as the first integer on line $i + 1$. Additionally, the second integer on line N must be equal to the first integer on line 1. Since there are many solutions, any one of them is acceptable.

Print a blank line between two successive test cases.

Sample Input

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

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

Sample Output

```
Case #1
some beads may be lost

Case #2
2 1
1 3
3 4
4 2
2 2
```