Alibaba Cup the 33rd ACM/ICPC Asia Hangzhou Regional Contest

Alibaba Cup

the 33rd ACM/ICPC Asia Regional Contest Hangzhou Site

The Problem Set

November 22~23, 2008

Problem A: A Pair of Graphs

Input File: a.in

Description

We say that two graphs are *equivalent* if and only if a one-to-one correspondence between their nodes can be established and under such a correspondence their edges are exactly the same. Given A and B, two undirected simple graphs with the same number of vertexes, you are to find a series of operations with the minimum cost that will make the two graphs equivalent. An operation may be one of the following two types:

a) Add an arbitrary edge (x, y), provided that (x, y) does not exist before such an operation. Such an operation costs I_A and I_B on two graphs, respectively.
b) Delete an existing edge (x, y), which costs D_A and D_B on two graphs, respectively.

Input

There are multiple test cases in the input file.

Each test case starts with three integers, N, M_A and M_B, ($1 \le N \le 8$, $0 \le M_A, M_B \le \frac{N*(N-1)}{2}$), the total number of vertexes, the number of edges in graph A, and the number of edges in graph B, respectively. Four integers I_A, I_B, D_A, and D_B come next, ($0 \le I_A, I_B, D_A, D_B \le 32767$), representing the costs as stated in the problem description. The next M_A + M_B lines describe the edges in graph A followed by those in graph B. Each line consists of exactly two integers, X and Y ($X \ne Y, 0 \le X, Y < N$).

Two successive test cases are separated by a blank line. A case with N = 0, M_A = 0, and M_B = 0 indicates the end of the input file, and should not be processed by your program.

Output

Please print the minimum cost in the format as illustrated below.

Sample Input	Output for the Sample Input
`1 0 0`	`Case #1: 0`
`1 2 3 7`	`Case #2: 1`
`4 2 3`	
`1 6 5 1`	

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

0 0 0
```

Problem B: Binary Integer

Input File: b.in

Description

An antique machine with $\binom{N}{3}$ switches capable of processing integers in the range $0..2^N - 1$ has just been discovered. Each switch is associated to a distinct integer in $0..2^N - 1$ with exactly three ones in its binary representation. By setting switches associated with number $X_0, X_1, \ldots, X_{M-1}$ to ***on****, any integer Y passing through the machine will render a result of $Y \oplus X_0 \oplus X_1 \oplus \ldots \oplus X_{M-1}$ (here "$\oplus$" stands for bitwise-XOR).*

We are interested in the number of configurations capable of transforming integer S into T with exactly K switches set to on. Could you write a program to help us?

Input

There are multiple test cases in the input file.

Each test case starts with two integers, N and K ($1 \le N \le 40, 0 \le K \le \min\{20, \binom{N}{3}\}$), followed by two binary integers, S and T, each containing exactly N bits.

Two successive test cases are separated by a blank line. A case with N = 0 and K = 0 indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print a single integer, the total number of ways to transform the first integer into the second one. Since the answer could be quite large, you are only required to find the result % 10007.

Sample Input

```
4 3
1101
1001

3 1
101
010

5 3
11010
10111
```

Output for the Sample Input

```
Case #1: 1
Case #2: 1
Case #3: 6
```

0 0	

Problem C: Cryptography Reloaded

Input File: c.in

Description

What do researchers working at ICPC (Institute for Cryptographic Programming and Computing) do for fun? Well, as you probably have expected, in addition to solving algorithm-related problems on online judges, they also like to toy with various cryptographic schemes. Recently one of the researchers, Tom, has become interested in RSA algorithm implementations used in handheld devices.

Note that the description of the general RSA algorithm is as follows:

*1) Choose two distinct prime numbers **p** and **q**, and let **n = pq**;*
*2) Choose an integer **e** such that $gcd(e,(p-1)(q-1)) = 1$;*
*3) Compute the integer **d** that satisfies the congruence relation $de \equiv 1(mod\ (p-1)(q-1))$.*

*Then, if person A wants to give person B a way to send an encrypted message to him, A can follow the above steps and release (**n**, **e**) as his public key. Upon receiving A's public key, B can simply encrypt message **x** ($0 \le x < n$) by computing $y = x^e \bmod n$. This would result in a message which ideally only A could decrypt with his private key **d**: $x = y^d \bmod n$.*

As the computation power of handheld devices is usually limited, a relatively small **e** is usually used to encrypt data. However this can lead to great security risks. For example, it is quite simple to recover p and q (i.e., factor n) when you have both the public key (n, e) and the private key d. Could you help Tom write a program to demonstrate this?

Input

There are multiple test cases in the input file.

Each test case contains three integers, n, d, and e ($n \le 10^{100}, 3 \le e \le 31$). All three integers are given without any preceding zeros. It is guaranteed that all numbers satisfy the condition as given in the problem statement.

Two successive test cases are separated by a blank line. A case with n = 0, d = 0 and e = 0 indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print two prime numbers, p and q, such that $n = pq$ and $p < q$, in the format as illustrated below.

Sample Input	Output for the Sample Input

55 27 3
 	
290203 168101 5
 	
0 0 0
	Case #1: 5 11
	Case #2: 29 10007

Problem D: Déjà vu

Input File: d.in

Description

An antique machine with $\binom{N}{3}$ switches capable of processing integers in the range $0..2^N - 1$ has just been discovered. Each switch is associated to a distinct integer in $0..2^N - 1$ with exactly three ones in its binary representation. By setting switches associated with number $X_0, X_1, \ldots, X_{M-1}$ to ***on****, any integer Y passing through the machine will render a result of $Y \oplus X_0 \oplus X_1 \oplus \ldots \oplus X_{M-1}$ (here "$\oplus$" stands for bitwise-XOR).*

Further inspections reveal that contrary to what we assumed in problem B, some of the switches on the machine are damaged due to their old age. We are interested in whether a configuration transforming integer S into T still exists, and if so, the minimum number of switches that have to be set to ***on*** to make it possible.

WARNING: a naïve algorithm might not be sufficient to solve this problem.

Input

There are multiple test cases in the input file.

Each test case starts with two integers, N and M ($1 \leq N \leq 20$), representing the number of bits and the number of functioning switches, respectively. Two integers, S and T ($0 \leq S, T < 2^N$), come in the next line, followed by another M lines, the ith one describing the value V_i associated to the ith switch ($0 \leq V_i < 2^N$).

Two successive test cases are separated by a blank line. A case with N = 0 and M = 0 indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print a single integer, the minimum number of operations, or "Impossible" (without quotes) if no feasible sequence exists.

Sample Input	Output for the Sample Input

`6 7`
`55 21`
`11`
`22`
`25`
	`Case #1: 2`
	`Case #2: Impossible`

56
41
49
28
 	
5 2
0 21
22
28
 	
0 0	

Problem E: Experiment on a … "Cable"

Input File: e.in

Description

The head technical person, Joey, at ACM (Association for Cyberspace Management) has just received a weird cable-like device - supposedly invented by programmers during a competition - for inspection.

The device may be viewed as a straight bi-directional cable, which can be used for transmitting arbitrary number of data packages simultaneously. The speed with which each package is sent will be pre-determined by the device and furthermore may vary within a certain range; however it will remain constant throughout each package's entire transmission process. The time it takes for a single data package with speed V to arrive at the opposite end of the cable is thus $\frac{L}{V}$, where L is the length of the cable.

In addition, the user may sometimes send a fixed-speed "detector" package, which is capable of reporting the number of data packages alongside itself at any time.

Finding this device highly amusing, Joey decides to perform an experiment on the odd "cable". He has scheduled N packages to be sent from the left side and another M to be sent from the other side of the cable; also, he has calculated the possible speed range for each data package. With this information in hand, Joey wants to estimate the *effectiveness* of a detector he will send. For a detector that departs at a certain time, its *effectiveness* can be represented as a real number in the range [0..1], which is simply the ratio of the time during which the detector has a chance of reporting all N + M packages (*explained below*) to the total time.

If Joey sends the detector at an arbitrary time in [S, T] with speed V from the left side of the cable, what is the *average* effectiveness he can achieve?

Note: For a detector to have a chance of reporting all N + M packages at time T_0, the device must be able to schedule all data packages with such speeds so that all can share the same position with the detector at time T_0.

Input

There are multiple test cases in the input file.

Each test case starts with one integer, L ($1 \le L \le 10^6$), the length of the cable. The next line contains one integer, N, the number of packages Joey will send from the left side, followed by N lines, the ith line with three real numbers, $MinV_i$, $MaxV_i$, and $Leave_i$ ($1 \le MinV_i \le MaxV_i$), which are the minimum speed, maximum speed, and departure time for package i, respectively. Another (M + 1) lines follow, describing the packages departing from the right side. The last line of the input contains three real numbers, S, T and V ($T - S \ge 1$), whose meanings are described above.

The total number of packages Joey sent will be in the interval $[1, 5000]$. It is guaranteed that the speed of any data packet, including that of the detector, will be no less than 0.01; also, all real numbers in the input will be given with at most two digits after the decimal point, and will belong to the interval: $[0, 10^6]$.

Two successive test cases are separated by a blank line. A case with L = 0 indicates the end of the input file, and should not be processed by your program.

Output

Please print the average effectiveness of the detector’s trip, with precision up to 0.00001.

Sample Input	Output for the Sample Input
`5`	`Case #1: 0.00000`
`1`	`Case #2: 0.25000`
`5.00 10.00 2.00`	
`2`	
`10.05 11.50 0.05`	
`1.68 2.00 0.01`	
`3.00 4.00 1000`	
`5`	
`1`	
`1.25 2.50 1.0`	
`0`	
`1.00 5.00 2.50`	
`0`	

Problem F: Fire-Control System

Input File: f.in

Description

A new mighty weapon has just been developed, which is so powerful that it can attack a sector of indefinite size, as long as the center of the circle containing the sector is the location of the weapon. We are interested in developing a fire-control system that calculates firing-solutions automatically.

The following example gives an example of a firing solution:

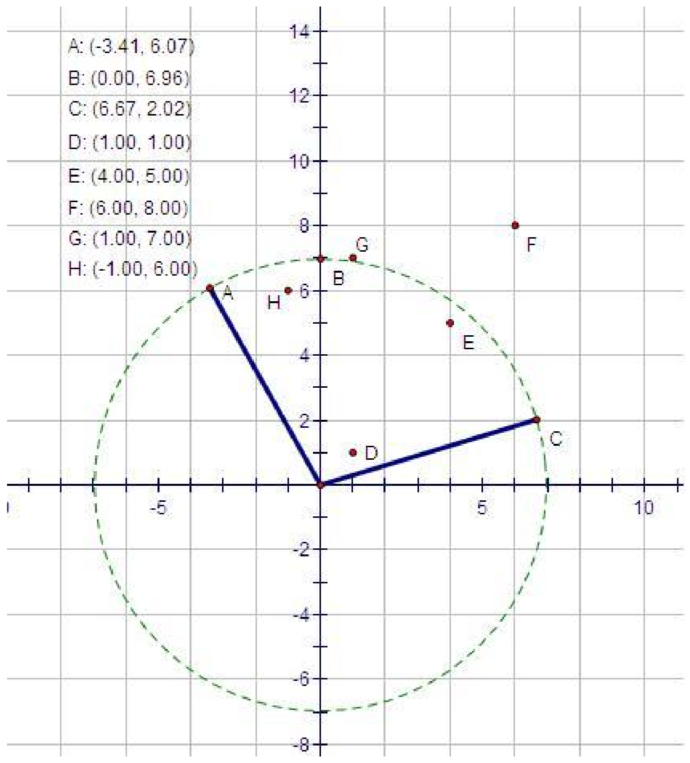

Figure 1

Here the firing region is the sector $\overline{ABC}$ *that covers six points: A, B, C, D, E, H.*

You may further assume that the weapon is always *located at point (0, 0), no targets will be on the point (0, 0) and the coordinates of the targets will be* distinct*.*

A firing solution is called *effective* if and only if it covers a minimum of K points out of N given points (targets) on the two-dimensional Cartesian plane. Furthermore, since the cost of a particular fire solution is in direct proportion to the size of the area it covers, a firing could be quite costly; thus we are only interested in the optimal firing solution with the minimum cost.

Input

There are multiple test cases in the input file.

Each test case starts with two non-negative integers, N and K ($1 \le N \le 5000, K \le N$), followed by N lines each containing two integers, X, and Y, describing the distinct location of one target. It is guaranteed that the absolute value of any integer does not exceed 1000.

Two successive test cases are separated by a blank line. A case with N = 0 and K = 0 indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print the required size (to two decimal places), in the format as indicated in the sample output.

Sample Input	Output for the Sample Input
`3 1`	`Case #1: 0.00`
`0 1`	`Case #2: 3.14`
`1 0`	
`-5 -6`	
`3 2`	
`0 2`	
`2 0`	
`-5 -6`	
`0 0`	

Problem G: Get-Together at Stockholm

Input File: g.in

Description

Peter has recently decided to hold a party at Stockholm, where the ACM/ICPC 2009 World Final will be held. Unfortunately, despite Peter's affluence, he is not able to invite all of his friends due to the astronomical price of the air ticket to Stockholm. He has devised the following rule to determine which subset of his friends will be invited:

a) Any invited person must have at least A acquaintances at the party. This is to ensure everyone at the party will not feel alien.
b) Any invited person must be unfamiliar with at least B people. Otherwise some people may not have the chance to communicate with a stranger.

Given the relationships between Peter's friends, you are to figure out whom Peter should invite to the party so as to maximize its size.

Input

There are multiple test cases in the input file.

Each test case starts with four integers, N, M, A, and B ($1 \leq N \leq 100, 0 \leq M \leq \frac{N*(N-1)}{2}, 0 \leq A, B \leq N-1$). Each of the following M lines contains two integers, X and Y, ($0 \leq X, Y \leq N-1, X \neq Y$), indicating that friend X and friend Y are acquaintances.

Two successive test cases are separated by a blank line. A case with N = 0, M = 0, A = 0 and B = 0 indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print a single integer, the maximum number of friends Peter will be able to invite.

Sample Input	Output for the Sample Input

`3 2 1 1`
`0 1`
`1 2`
	`Case #1: 0`
	`Case #2: 4`

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

Problem H: History of Languages

Input File: h.in

Description

We are examining two specific classes of languages (a possibly infinite set of strings) in this problem. Fortunately (or maybe unfortunately), we are not given the strings contained in each language directly, rather we are given two deterministic finite automatons that describe such languages.

Your task here is simple: to determine if the languages described by the two automatons are the same.

For those of you who are unfamiliar with the concept of deterministic finite automata, you may use the following information:

A deterministic finite automata (DFA) is a 5-tuple, (S, Σ, T, s, A), consisting of

- *a finite set of states (S)*
- *a finite set called the alphabet (Σ)*
- *a transition function ($T : S \times \Sigma \to S$)*
- *a start state ($s \in S$)*
- *a set of accept states ($A \subseteq S$)*

*Let **M** be a DFA such that **M** = (S, Σ, T, s, A), and $X = x_0x_1 \ldots x_n$ be a string over the alphabet Σ. **M** accepts the string X if a sequence of states, $r_0, r_1, \ldots, r_n$, exists in S with the following conditions:*

1. $r_0 = s$
2. $r_{i+1} = T(r_i, x_i)$, *for* $i = 0, \ldots, n-1$
3. $r_n \in A$.

- *Wikipedia*

Input

There are multiple test cases in the input file.

The first line of each test case is one integer, T ($2 \leq T \leq 26$), the size of the alphabet. In each test case, the description of automaton A comes before that of automaton B. The description of each automaton starts with one line containing N ($1 \leq N \leq 2000$), the number of states in the automaton, followed by N lines, each line of the format: F, X_0, X_1, …, X_{T-1}, $(F \in \{0,1\}, -1 \le X_i < N)$. If F = 1, then the state is an accepting state; also, if $X_i = -1$, it means that the state has no corresponding transition available for character i. The start state of both finite automatons will always be state 0.

Two successive test cases are separated by a blank line. A case with a single 0 indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print “Yes” if the two languages described by the automatons are equivalent; output “No” otherwise.

Sample Input	Output for the Sample Input

2
3
1 -1 1
0 -1 2
0 -1 0
2
1 -1 1
0 -1 0
 	
3
4
1 -1 -1 1
1 -1 -1 2
1 -1 -1 3
1 -1 -1 1
2
1 -1 -1 1
1 -1 -1 0
 	
0
	Case #1: No
	Case #2: Yes

Problem I: Image Processing

Input File: i.in

Description

According to Wikipedia, image processing is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it.

The task you are facing here is a relatively easy one (compared to our general conception of image processing!). Given a black-and-white image of size R * C with some digits (and possibly other shapes) on it, your program needs to figure out the digits written on the image. Specifically, the digits drawn on the graph will adhere to the following rules:

1) Digits are drawn with a series of strokes. A stroke can be regarded as a rectangle of any size on the image, and its edges will always be parallel to either x-axis or y-axis. The number of strokes required to draw each digit will be exactly as follows:

0	1	2	3	4	5	6	7	8	9
4	1	5	4	3	5	5	2	5	5

   Refer to Figure 2 if you're unclear about how the digits are drawn.
2) Although the width of strokes used to draw a digit might be different, the outer shapes of digits will strictly follow those specified in Figure 2.
3) In order for a digit to be recognizable, all parts (strokes and joints) presented in the graph below must also be clearly distinguishable in the image.
   (Refer to the last sample test case if you are unsure about this requirement; in that test case, when the middle stroke of 2 is omitted, the number should not be considered as recognizable.)
4) You may assume that the image is not rotated, and there is no noise in the input.

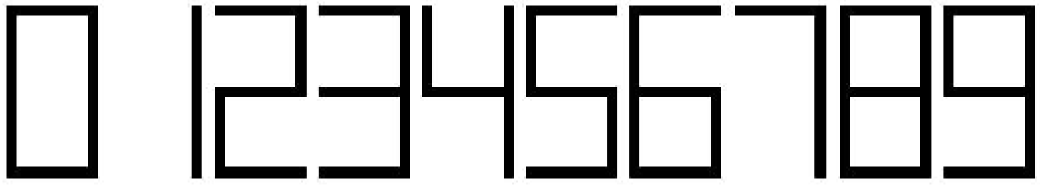

Figure 2

Please output the sum of digits recognizable in the graph. In the case that no characters is recognizable, please output 0 instead.

Input

There are multiple test cases in the input file.

Each test case starts with two integers, R and C ($1 \le R, C \le 500$), specifying the number of rows / columns of the graph. Each of the following R lines contains consecutive C characters ('0' or '1'), describing the image to be processed.

Two successive test cases are separated by a blank line. A case with R = 0, C = 0 indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print a single integer, the sum of recognizable numbers.

Sample Input	Output for the Sample Input

`5 12`
`001101011111`
`000101000011`
`000101001111`
`001101000011`
`000000000111`
 	
`5 3`
`111`
`010`
`110`
`010`
`110`
 	
`6 14`
`11111000011111`
`11001000000011`
`11111001000000`
`11111001001110`
`11001011001010`
`11111000001110`
 	
`5 2`
`11`
	`Case #1: 4`
	`Case #2: 0`
	`Case #3: 15`
	`Case #4: 3`
	`Case #5: 2`

```
01
11
01
11

6 9
111100111
000100001
000100011
011100010
010000011
011110000

0 0
```

Problem J: Junk-Mail Filter

Input File: j.in

Description

Recognizing junk mails is a tough task. The method used here consists of two steps:

1) Extract the common characteristics from the incoming email.
2) Use a filter matching the set of common characteristics extracted to determine whether the email is a spam.

We want to extract the set of common characteristics from the N sample junk emails available at the moment, and thus having a handy data-analyzing tool would be helpful. The tool should support the following kinds of operations:

a) “M X Y”, meaning that we think that the characteristics of spam X and Y are the same. Note that the relationship defined here is *transitive*, so relationships (other than the one between X and Y) need to be created if they are not present at the moment.
b) “S X”, meaning that we think spam X had been misidentified. Your tool should remove all relationships that spam X has when this command is received; after that, spam X will become an isolated node in the relationship graph.

Initially no relationships exist between any pair of the junk emails, so the number of distinct characteristics at that time is N.

Please help us keep track of any necessary information to solve our problem.

Input

There are multiple test cases in the input file.

Each test case starts with two integers, N and M ($1 \le N \le 10^5$, $0 \le M \le 10^6$), the number of email samples and the number of operations. M lines follow, each line is one of the two formats described above.

Two successive test cases are separated by a blank line. A case with N = 0 and M = 0 indicates the end of the input file, and should not be processed by your program.

Output

For each test case, please print a single integer, the number of distinct common characteristics, to the console. Follow the format as indicated in the sample below.

Sample Input	Output for the Sample Input
`5 6`	`Case #1: 3`
`M 0 1`	`Case #2: 2`
`M 1 2`	
`M 1 3`	
`S 1`	
`M 1 2`	
`S 3`	
`3 1`	
`M 1 2`	
`0 0`	