The 2012 ACM-ICPC Asia Regional Contest
Chengdu Site

sponsored by IBM & Huawei
hosted by Chengdu Neusoft University
Chengdu, China

November 11, 2012

This problem set should contain eleven (11) problems on eighteen (18) pages. Please inform a runner immediately if something is missing from your problem set.

Problem A. Browsing History

Description

One day when you are going to clear all your browsing history, you come up with an idea: You want to figure out what your most valued site is. Every site is given a value which equals to the sum of ASCII values of all characters in the URL. For example `aa.cc` has value of 438 because $438 = 97 + 97 + 46 + 99 + 99$. You just need to print the largest value amongst all values of sites.

Things are simplified because you found that all entries in your browsing history are of the following format: [domain], where [domain] consists of lower-case Latin letters and "." only. See the sample input for more details.

Input

There are several test cases.

For each test case, the first line contains an integer n ($1 \le n \le 100$), the number of entries in your browsing history.

Then follows n lines, each consisting of one URL whose length will not exceed 100.

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is a number indicating the desired answer.

Sample input and output

Sample Input	Sample Output

`1`
`aa.cc`
`2`
`www.google.com`
`www.wikipedia.org`
	`Case 1: 438`
	`Case 2: 1728`

Problem B. Candy

Description

LazyChild is a lazy child who likes candy very much. Despite being very young, he has two large candy boxes, each contains n candies initially. Everyday he chooses one box and open it. He chooses the first box with probability p and the second box with probability $(1-p)$. For the chosen box, if there are still candies in it, he eats one of them; otherwise, he will be sad and then open the other box.

He has been eating one candy a day for several days. But one day, when opening a box, he finds no candy left. Before opening the other box, he wants to know the expected number of candies left in the other box. Can you help him?

Input

There are several test cases.

For each test case, there is a single line containing an integer n ($1 \le n \le 2 \times 10^5$) and a real number p ($0 \le p \le 1$, with 6 digits after the decimal).

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is a real number indicating the desired answer.

Any answer with an absolute error less than or equal to 10^{-4} would be accepted.

Sample input and output

Sample Input	Sample Output
`10 0.400000`	`Case 1: 3.528175`
`100 0.500000`	`Case 2: 10.326044`
`124 0.432650`	`Case 3: 28.861945`
`325 0.325100`	`Case 4: 167.965476`
`532 0.487520`	`Case 5: 32.601816`
`2276 0.720000`	`Case 6: 1390.500000`

Problem C. Triangle

Description

You have a piece of iron wire with length of n unit. Now you decide to cut it into several *ordered* pieces and fold each piece into a triangle satisfying:

- All triangles are *integral.*
- All triangles are pairwise *similar*.

You should count the number of different approaches to form triangles. Two approaches are considered different if either of the following conditions is satisfied:

- They produce different numbers of triangles.
- There exists i that the i^{th} (again, pieces are ordered) triangle in one approaches is not *congruent* to i^{th} triangle in another plan.

The following information can be helpful in understanding this problem.

- A triangle is *integral* when all sides are integer.
- Two triangles are *congruent* when all corresponding sides and interior angles are equal.
- Two triangles are *similar* if they have the same shape, but can be different sizes.
- For $n = 9$ you have 6 different approaches to do so, namely

$$\begin{array}{lll} (1,1,1) & (1,1,1) & (1,1,1) \\ (1,1,1) & (2,2,2) & \\ (2,2,2) & (1,1,1) & \\ (1,4,4) & & \\ (2,3,4) & & \\ (3,3,3) & & \end{array}$$

where (a, b, c) represents a triangle with three sides a, b, c.

Input

There are several test cases.

For each test case there is a single line containing one integer n ($1 \leq n \leq 5 * 10^6$).

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is the number of approaches, moduled by $10^9 + 7$.

Sample input and output

Sample Input	Sample Output
1	Case 1: 0
2	Case 2: 0
3	Case 3: 1
4	Case 4: 0
5	Case 5: 1
6	Case 6: 2
8	Case 7: 1
9	Case 8: 6
10	Case 9: 3
11	Case 10: 4
12	Case 11: 10
15	Case 12: 25
19	Case 13: 10
20	Case 14: 16
100	Case 15: 525236
1000	Case 16: 523080925

Problem D. Graph

Description

P. T. Tigris is a student currently studying graph theory. One day, when he was studying hard, GS appeared around the corner shyly and came up with a problem:

Given a graph with n nodes and m undirected weighted edges, every node having one of two colors, namely black (denoted as 0) and white (denoted as 1), you're to maintain q operations of either kind:

- `Change` x: Change the color of x^{th} node. A black node should be changed into white one and vice versa.
- `Asksum` A B: Find the sum of weight of those edges whose two end points are in color A and B respectively. A and B can be either 0 or 1.

P. T. Tigris doesn't know how to solve this problem, so he turns to you for help.

Input

There are several test cases.

For each test case, the first line contains two integers, n and m ($1 \leq n, m \leq 10^5$), where n is the number of nodes and m is the number of edges.

The second line consists of n integers, the i^{th} of which represents the color of the i^{th} node: 0 for black and 1 for white.

The following m lines represent edges. Each line has three integer u, v and w, indicating there is an edge of weight w ($1 \leq w \leq 2^{31} - 1$) between u and v ($u \neq v$).

The next line contains only one integer q ($1 \leq q \leq 10^5$), the number of operations.

Each of the following q lines describes an operation mentioned before.

Input is terminated by EOF.

Output

For each test case, output several lines.

The first line contains "Case **X**:", where **X** is the test case number (starting from 1).

And then, for each "`Asksum`" query, output one line containing the desired answer.

Sample input and output

Sample Input	Sample Output
4 3 0 0 0 0 1 2 1 2 3 2 3 4 3 4 Asksum 0 0 Change 2 Asksum 0 0 Asksum 0 1 4 3 0 1 0 0 1 2 1 2 3 2 3 4 3 4 Asksum 0 0 Change 3 Asksum 0 0 Asksum 0 1	Case 1: 6 3 3 Case 2: 3 0 4

Problem E. Spy

Description

"Be subtle! Be subtle! And use your spies for every kind of business. "

— Sun Tzu

"A spy with insufficient ability really sucks"

— An anonymous general who lost the war

You, a general, following Sun Tzu's instruction, make heavy use of spies and agents to gain information secretly in order to win the war (and return home to get married, what a flag you set up). However, the so-called "secret message" brought back by your spy, is in fact encrypted, forcing yourself into making deep study of message encryption employed by your enemy.

Finally you found how your enemy encrypts message. The original message, namely s, consists of lowercase Latin alphabets. Then the following steps would be taken:

- Step 1: Let $r = s$
- Step 2: Remove r's suffix (may be empty) whose length is less than length of s and append s to r. More precisely, firstly donate $r[1..n]$, $s[1..m]$, then an integer i is chosen, satisfying $i \leq n$, $n - i < m$, and we make our new $r = r[1..i] + s[1..m]$. This step might be taken for several times or not be taken at all.

What your spy brought back is the encrypted message r, you should solve for the minimal possible length of s (which is enough for your tactical actions).

Input

There are several test cases.

For each test case there is a single line containing only one string r (The length of r does not exceed 10^5). You may assume that the input contains no more than 2×10^6 characters.

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is the desired answer.

Sample input and output

Sample Input	Sample Output
`abc`	`Case 1: 3`
`aab`	`Case 2: 2`
`abcadabcabcad`	`Case 3: 5`
`aaabbbaaaabbbaa`	`Case 4: 6`
`abcababcd`	`Case 5: 4`

Problem F. Age of Empires

Description

P. T. Tigris has been an avid computer game lover since he was a little boy fifteen years ago. “Age of Empires” was one of the most famous real-time strategy games in that past years. Resource is one of the most important consideration when playing this game. In this game, there are four different types of resources: food, wood, stone and gold.

Food is the most important resource in the game, which is used for producing many military units and researching important technologies. Wood is most often required for buildings or training ranged units. Stone is mainly used to build static defenses like Towers and Walls, but it is also used for some units and technologies as well as Wonders. Finally, gold is used for creating most units and upgrades and is a precious resource which becomes more important as the game progresses.

The Villager is a common civilian unit for almost every game. They are the backbone of all civilizations. The Villagers are arguably the most important units in the game because they are able to collect all the resources.

P. T. Tigris is very interesting in the optimal strategy of game playing. He wants to discover the fastest way to gather enough resources. We may assume that P. T. Tigris has N villagers at the beginning of the game with initially no food, wood, stone or gold at all. P. T. Tigris knows that every second, each villager could gather A_1 units of food, or B_1 units of wood, or C_1 units of stone, or D_1 units of gold. Note that the villager can not split one second into smaller pieces to gather different types of resource. For example, a single villager can not gather $A_1/2$ units of food and $B_1/2$ units of wood for a single second. Moreover, all kinds of recourse are gathered exactly the end of that second.

Different villagers could gather different types of resources at a time. P. T. Tigris should decide which type of resources he should gather for each villager at every second, so that he could have A_2 units of food and B_2 units of wood and C_2 units of stone and D_2 units of gold as soon as possible.

P. T. Tigris could also training more villagers to speed up his process of gathering. To training a villager, P. T. Tigris should spend X units of food at the beginning of a second, and a new villager will able to work after T seconds. Please note that at he beginning of the second you start training a villager, you must have not less than X units of food. All villagers are trained at the Town Center but unfortunately there is only one Town Center for P. T. Tigris. So he can only train one villager at a time.

It is really difficult to find the optimal answer. P. T. Tigris decides to ask you to write a program to help him calculate the minimum time to get the required amount of resources.

Input

There are several test cases.

For each test case, the first line contains four integers A_1, B_1, C_1 and D_1 ($1 \leq A_1$, B_1, C_1, $D_1 \leq 10^{18}$), indicating the amount of resource a villager can gather for each type in a second.

The second line also contains four integers A_2, B_2, C_2 and D_2 ($0 \leq A_2$, B_2, C_2, $D_2 \leq 10^{18}$), denoting the amount of resource P. T. Tigris want to have.

The third line contains three integers N, X and T ($1 \leq N, X, T \leq 10^5$), denoting that P. T. Tigris has N villager at the beginning of the game, and it will spend X units of food and T seconds to train each new villager.

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is an integer indicating the minimum time required to reach his goal.

Sample input and output

Sample Input	Sample Output

`1 1 1 1`
`1 1 1 1`
`4 1 1`
`1 1 1 1`
`2 2 2 2`
`1 1 1`
`1 1 1 1`
`10 10 10 10`
`1 1 25`
	`Case 1: 1`
	`Case 2: 5`
	`Case 3: 34`

Problem G. Blackjack

Description

You, as a great mathematician and a former member of Blackjack Team, are recently declared "unwelcome person" by managers of local casinos. Extremely bored, you reminded of your life forming a team beating casinos at blackjack worldwide, and decided to help your friends in winning blackjack games.

Blackjack, also known as twenty-one, is a game frequently seen in casinos, played with one deck, or several decks of 52 cards. The version your friend plays is slightly different from what we used to see in usual casinos. In this version, the game is played between a player and a dealer with a deck of n cards, namely $a_1, a_2, \ldots, a_n$, instead of regular decks of 52 cards in standard version. The i^{th} card has the unique numeric value a_i, which is important in following description of rules.

The game is played in several rounds as long as not less than k ($k \geq 10$) cards left in the deck. Cards are dealt from a_1 to a_n, while each card is dealt out at most once. In each round, the player is dealt one card, then the dealer, then the player, then dealer. They now have two cards in their hand respectively. Then the player would keep on taking a hit until he busts (total value of his hand exceeds 21 points) or he feels it's enough (total value of his hand exceeds 15 points) or he has taken 3 hits already. He immediately loses the round if he busts. If he has taken 3 hits without bust, making his hand consist of 5 cards, he wins the round, ending the round right away. Then the dealer will use the exactly same strategy as the player. Of course, the dealer loses the round immediately if he busts, wins the round at once if his hand consists of 5 cards, with the same rule applying. If after taking hits neither the player nor the dealer wins or loses, sums of points (described below) in their hands will be compared, and the person with larger one will win the round. In case of tie, neither wins or loses. Of course, this ends the current round.

In the casino your friend plays, there is a special rule: before the game starts, the player is required to cut the deck of card exactly once. By saying cut the deck we mean to change deck of cards from

$$a_1, a_2, \ldots, a_n$$

to

$$a_p, a_{p+1}, \ldots, a_q, a_1, a_2, \ldots, a_{p-1}, a_{q+1}, \ldots, a_n (1 < p \leq q < n)$$

With your super power (in hacking) you now know the deck of cards to play. Now you want to instruct your friend to cut the cards by telling your friend p and q in a secret manner, in order to maximize number of rounds he wins.

Input

There are several test cases.

For each test case, the first line contains two integers, namely n ($20 \leq n \leq 2000$) and k ($10 \leq k \leq n$).

The following lines contain totally n characters separated by spaces or line breaks. Characters can be any of A,2,3,4,5,6,7,8,T,J,Q,K where A stands for numeric 1 and T,J,Q,K stands for numeric 10.

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is a number indicating the desired answer.

Sample input and output

<table>
<tr><th>Sample Input</th><th>Sample Output</th></tr>
<tr><td><pre>20 10
8 4 7 8 8 K 5 A Q Q A Q 6 4 J 6 9 5
3 9
40 10
3 J 7 7 2 T J 6 A 4 4 8 J T 6 A 6 2 K 9
6 5 7 J T 3 5 5 3 7 7 J 5 3 A 5 9 Q 6 7</pre></td><td><pre>Case 1: 3
Case 2: 6</pre></td></tr>
</table>

Problem H. Homework

Description

GS is suffering from tons of boring math assignment. He find it make him tired and impatient so he asks you to finish his assignment in hope that he could hang out in many places of interest and enjoy his life.

In this assignment, you're asked to solve the following problem:

Given a recurrent function

$$f[n] = \sum_{i=1}^{t} c[i]f[n-i]$$

and boundary values

$$f[i], 1 \leq i \leq m$$

You should solve for $f[n]$.

What a easy problems! Wait for a moment, you see a few lines in the last paragraph. It reads as follows: To make the problem a little hard, you are now informed that, at some special values of n (there are q such values, namely $n_1, n_2, \ldots, n_q$), the recurrent formula changes into something else, which means for the k^{th} such value n_k, the recurrent formula changes into

$$f[n_k] = \sum_{i=1}^{t_k} c_k[i]f[n_k - i]$$

Still an easy problem, isn't it?

Since $f[n]$ may be quite large, you just need to output $f[n]$ module $10^9 + 7$.

Input

There are several test cases.

For each test case, the first line contains three integers n ($m < n \leq 10^9$), m ($1 \leq m \leq 100$), q ($0 \leq q \leq 100$). The second line contains m integers, namely $f[1], f[2], \ldots, f[m]$.

The following line contains several integers, first comes t ($t \leq 100$), then t integers namely $c[1]$, $c[2]$, …, $c[t]$.

The following q lines describe q special cases of the recurrent formula, each containing several integers, namely n_k, t_k ($t_k \leq 100$, $t_k < n_k$), $c_k[1], c_k[2], \ldots, c_k[t_k]$, as mentioned earlier. It is satisfied that $n_i \neq n_j$ if $i \neq j$.

All integers are non-negative. Unless specified, all integers are not greater than 10^9.

Input is terminated by EOF.

You might assume that all given data is correct.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is the desired answer.

Sample input and output

Sample Input	Sample Output

`7 5 0`
`1 1 2 3 5`
`2 1 1`
`10 5 1`
`1 1 2 3 5`
`2 1 1`
`10 2 1 2`
	`Case 1: 13`
	`Case 2: 76`

Sample explanation

In the first sample, you are to solve for $f[7]$ where $f[n] = f[n-1] + f[n-2]$ and $f[1] = 1$, $f[2] = 1$, $f[3] = 2$, $f[4] = 3$, $f[5] = 5$.

In the second example, you are to solve for $f[10]$ where $f[n] = f[n-1] + f[n-2]$ and $f[1] = 1$, $f[2] = 1$, $f[3] = 2$, $f[4] = 3$, $f[5] = 5$, as well as specially $f[10] = f[9] + 2f[8]$.

Problem I. Count

Description

Prof. Tigris is the head of an archaeological team who is currently in charge of an excavation in a site of ancient relics.

This site contains relics of a village where civilization once flourished. One night, examining a writing record, you find some text meaningful to you. It reads as follows.

"Our village is of glory and harmony. Our relationships are constructed in such a way that everyone except the village headman has exactly one direct boss and nobody will be the boss of himself, the boss of boss of himself, etc. Everyone expect the headman is considered as his boss's subordinate. We call it relationship configuration. The village headman is at level 0, his subordinates are at level 1, and his subordinates' subordinates are at level 2, etc. Our relationship configuration is harmonious because all people at same level have the same number of subordinates. Therefore our relationship is …"

The record ends here. Prof. Tigris now wonder how many different harmonious relationship configurations can exist. He only cares about the holistic shape of configuration, so two configurations are considered identical if and only if there's a bijection of n people that transforms one configuration into another one.

Please see the illustrations below for explanation when $n = 2$ and $n = 4$.

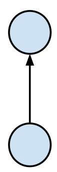

Figure 1: $n = 2$

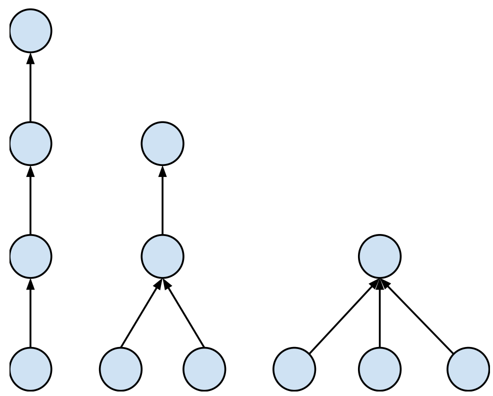

Figure 2: $n = 4$

The result might be very large, so you should take module operation with modules $10^9 + 7$ before print your answer.

Input

There are several test cases.

For each test case there is a single line containing only one integer n ($1 \leq n \leq 1000$).

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is the desired answer.

Sample input and output

Sample Input	Sample Output
1 2 3 40 50 600 700	Case 1: 1 Case 2: 1 Case 3: 2 Case 4: 924 Case 5: 1998 Case 6: 315478277 Case 7: 825219749

Problem J. Exam

Description

Rikka is a high school girl suffering seriously from Chūnibyō (the age of fourteen would either act like a know-it-all adult, or thinks they have special powers no one else has. You might google it for detailed explanation) who, unfortunately, performs badly at math courses. After scoring so poorly on her maths test, she is faced with the situation that her club would be disband if her scores keeps low.

Believe it or not, in the next exam she faces a hard problem described as follows.

Let's denote $f(x)$ number of ordered pairs satisfying $(a * b)|x$ (that is, $x \bmod (a * b) = 0$) where a and b are positive integers. Given a positive integer n, Rikka is required to solve for $f(1) + f(2) + \ldots + f(n)$.

According to story development we know that Rikka scores slightly higher than average, meaning she must have solved this problem. So, how does she manage to do so?

Input

There are several test cases.

For each test case, there is a single line containing only one integer n $(1 \leq n \leq 10^{11})$.

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) and **Y** is the desired answer.

Sample input and output

Sample Input	Sample Output
1 3 6 10 15 21 28	Case 1: 1 Case 2: 7 Case 3: 25 Case 4: 53 Case 5: 95 Case 6: 161 Case 7: 246

Problem K. Yet Another Multiple Problem

Description

There are tons of problems about integer multiples. Despite the fact that the topic is not original, the content is highly challenging. That's why we call it "Yet Another Multiple Problem".

In this problem, you're asked to solve the following question: Given a positive integer n and m decimal digits, what is the minimal positive multiple of n whose decimal notation does not contain any of the given digits?

Input

There are several test cases.

For each test case, there are two lines. The first line contains two integers n and m ($1 \leq n \leq 10^4$). The second line contains m decimal digits separated by spaces.

Input is terminated by EOF.

Output

For each test case, output one line "Case **X**: **Y**" where **X** is the test case number (starting from 1) while **Y** is the minimal multiple satisfying the above-mentioned conditions or "-1" (without quotation marks) in case there does not exist such a multiple.

Sample input and output

Sample Input	Sample Output

`2345 3`
`7 8 9`
`100 1`
`0`
	`Case 1: 2345`
	`Case 2: -1`