Pacific Northwest Region Programming Contest

Brigham Young University, Hawaii
Eastern Washington University
George Fox University
University of British Columbia
University of California, Santa Cruz
University of Puget Sound

November 2nd, 2013

Reminders

- For all problems, read the input data from standard input and write the results to standard output.
- In general, when there is more than one integer or word on an input line, they will be separated from each other by exactly one space. No input lines will have leading or trailing spaces, and tabs will never appear in any input.
- Compiler options are as follows:

```
g++ -g -O2 -std=gnu++0x -static $*
gcc -g -O2 -std=gnu99 -static $* -lm
javac -encoding UTF-8 -sourcepath . -d . $* runjava
java -client -Xss8m -Xmx1024m $*
```

Problem A — limit 5 seconds

Assignments

When Starfleet headquarters gets a request for an exploration expedition, they need to determine which ship from those currently docked in the docking bay to send. They decide to send whichever ship is currently able to make the expedition based on how much fuel is currently stored on the ship as well as how long it will take the ship to arrive at the expected destination. Due to the age and current maintenance of the ships, each ship travels at a different top speed and has a different fuel consumption rate. Each ship reaches its top speed instantaneously.

Input

Input begins with a line with one integer T $(1 \le T \le 50)$ denoting the number of test cases. Each test case begins with a line with two space-separated integers N and D, where N $(1 \le N \le 100)$ denotes the number of ships in the docking bay and D $(1 \le D \le 10^6)$ denotes the distance in light-years to the expedition site. Next follow N lines with three space-separated integers v_i, f_i, and c_i, where v_i $(1 \le v_i \le 1000)$ denotes the top speed of ship i in light-years per hour, f_i $(1 \le f_i \le 1000)$ denotes the fuel on ship i in kilos of deuterium, and c_i $(1 \le c_i \le 1000)$ denotes the fuel consumption of ship i in kilos of deuterium per hour.

Output

For each test case, print a single integer on its own line denoting the number of ships capable of reaching the expedition site. Be careful with integer division!

Sample Input	**Sample Output**

```
2
3 100
52 75 10
88 13 44
56 9 5
2 920368
950 950 1
943 976 1
```
	```
	2
	1
	```

Problem B — limit 5 seconds

Bones's Battery

Bones is investigating what electric shuttle is appropriate for his mom's school district vehicle. Each school has a charging station. It is important that a trip from one school to any other be completed with no more than K rechargings. The car is initially at zero battery and must always be recharged at the start of each trip; this counts as one of the K rechargings. There is at most one road between each pair of schools, and there is at least one path of roads connecting each pair of schools. Given the layout of these roads and K, compute the necessary range required of the electric shuttle.

Input

Input begins with a line with one integer T ($1 \le T \le 50$) denoting the number of test cases. Each test case begins with a line containing three integers N, K, and M ($2 \le N \le 100$, $1 \le K \le 100$), where N denotes the number of schools, K denotes the maximum number of rechargings permitted per trip, and M denotes the number of roads. Next follow M lines each with three integers u_i, v_i, and d_i ($0 \le u_i, v_i < N$, $u_i \ne v_i$, $1 \le d_i \le 10^9$) indicating that road i connects schools u_i and v_i (0-indexed) bidirectionally with distance d_i.

Output

For each test case, output one line containing the minimum range required.

Sample Input	Sample Output

```
2
4 2 4
0 1 100
1 2 200
2 3 300
3 0 400
10 2 15
0 1 113
1 2 314
2 3 271
3 4 141
4 0 173
5 7 235
7 9 979
9 6 402
6 8 431
8 5 462
0 5 411
1 6 855
2 7 921
3 8 355
4 9 113
```
	```
	300
	688
	```

Problem C — limit 20 seconds

Crusher's Code

Wesley Crusher is the teaching assistant for Introduction to Algorithms. During his first class, the cadets were asked to come up with their own sorting algorithms. Monty came up with the following code:

```
while (!sorted(a)) {
   int i = random(n) ;
   int j = random(n) ;
   if (a[min(i,j)] > a[max(i,j)])
      swap(a[i], a[j]) ;
}
```

Carlos, inspired, came up with the following code:

```
while (!sorted(a)) {
   int i = random(n-1) ;
   int j = i + 1 ;
   if (a[i] > a[j])
      swap(a[i], a[j]) ;
}
```

Wesley needs to determine which algorithm is better.

For a given input array of up to 8 values, calculate and print the expected number of iterations for each algorithm. That is, on average, how many iterations should each algorithm take for the given input?

Input

The first line contains T, the number of test cases: $2 \le T \le 100$.

Each test case is given on a single line. The first value is N, the number of array elements; $2 \le N \le 8$. This is followed on the same line by N integer array elements. The array elements will have values between 0 and 100 inclusive. The array elements may not be distinct.

Output

For each test case, print out the expected number of iterations for Monty's algorithm and for Carlos's algorithm, as shown in the sample output section. There should be exactly one space between words and no spaces at the start of each line or at the end of each line. There should be exactly six digits after the decimal point. Rounding should be to nearest representable value.

Sample Input	Sample Output

```
12
2 1 2
2 2 1
3 1 2 3
3 3 2 1
4 1 2 3 4
4 4 3 2 1
4 2 1 4 3
5 1 1 1 1 1
5 5 4 3 2 1
8 8 7 6 5 4 3 2 1
8 3 1 4 1 5 9 2 6
8 2 7 1 8 2 8 1 8
```
	```
	Monty 0.000000 Carlos 0.000000
	Monty 2.000000 Carlos 1.000000
	Monty 0.000000 Carlos 0.000000
	Monty 6.000000 Carlos 5.000000
	Monty 0.000000 Carlos 0.000000
	Monty 14.666667 Carlos 12.500000
	Monty 12.000000 Carlos 4.500000
	Monty 0.000000 Carlos 0.000000
	Monty 26.382275 Carlos 23.641975
	Monty 89.576273 Carlos 79.496510
	Monty 79.161905 Carlos 33.422840
	Monty 63.815873 Carlos 38.910494
	```

Problem D — limit 10 seconds

Delta Quadrant

Much of the Delta Quadrant remains unexplored, but it is known that there are many dangerous regions inhabited by warring races. Yet, there is a small number of declared neutral zones that reach from planet to planet in which it is safe to travel.

A critical summit is planned, but it is on hold until a quorum of its members is attained. To reach this quorum, almost all of the delegates from the Delta Quadrant must be fetched and brought to the same location, where the Enterprise can pick them up for the summit.

The Enterprise is on its way to the Delta Quadrant. You must find the fastest way, using a single ship leaving from any one of the planets in the Delta Quadrant and returning to that same planet, to visit all but a given number of planets.

There are N planets and $N - 1$ safe paths through neutral zones connecting the planets. Each path has a known time duration required for traversal. Find the shortest time required, starting from an arbitrary planet, to visit $N - k$ planets, and return.

Input

The first line contains T, the number of problems; $1 \leq T \leq 50$. Each problem instance starts with a line with two integers, N, the number of planets, and k, the count of planets that need not be visited; $2 \leq N \leq 10,000$ and $0 \leq k \leq \min(N - 1, 20)$. Following that are $N - 1$ lines, each containing three integers, which are, in order, the source and destination planet numbers (from 0 to $N - 1$) and then the time required to traverse that path. The time is between 0 and $1,000,000$, inclusive. The input graph will be connected.

Output

For each problem instance, print a single integer representing the final distance of travel required.

Sample Input

```
3
2 0
0 1 3000
4 1
0 1 81
1 2 41
2 3 59
9 2
0 1 1000
1 2 1200
0 3 1000
3 4 1200
0 5 1000
5 6 1200
0 7 1800
7 8 600
```

Sample Output

```
6000
200
13200
```

Problem E — limit 20 seconds

Enterprising Escape

The Enterprise is surrounded by Klingons! Find the escape route that has the quickest exit time, and print that time.

Input is a rectangular grid; each grid square either has the Enterprise or some class of a Klingon warship. Associated with each class of Klingon warship is a time that it takes for the Enterprise to defeat that Klingon. To escape, the Enterprise must defeat each Klingon on some path to the perimeter. Squares are connected by their edges, not by corners (thus, four neighbors).

Input

The first line will contain T, the number of cases; $2 \leq T \leq 100$. Each case will start with line containing three numbers k, w, and h. The value for k is the number of different Klingon classes and will be between 1 and 25, inclusive. The value for w is the width of the grid and will be between 1 and 1000, inclusive. The value for h is the height of the grid and will be between 1 and 1000, inclusive.

Following that will be k lines. Each will consist of a capital letter used to label the class of Klingon ships followed by the duration required to defeat that class of Klingon. The label will not be "`E`". The duration is in minutes and will be between 0 and 100,000, inclusive. Each label will be distinct.

Following that will be h lines. Each will consist of w capital letters (with no spaces between them). There will be exactly one "`E`" across all h lines, denoting the location of the Enterprise; all other capital letters will be one of the k labels given above, denoting the class of Klingon warship in the square.

Output

Your output should be a single integer value indicating the time required for the Enterprise to escape.

Sample Input

```
2
6 3 3
A 1
B 2
C 3
D 4
F 5
G 6
ABC
FEC
DBG
2 6 3
A 100
B 1000
BBBBBB
AAAAEB
BBBBBB
```

Sample Output

```
2
400
```

Problem F — limit 5 seconds

Federation Favorites

En route to Rigel 7, Chief Engineer Geordi Laforge and Data were discussing favorite numbers. Geordi exclaimed he preferred Narcissistic Numbers: those numbers whose value is the same as the sum of the digits of that number, where each digit is raised to the power of the number of digits in the number.

Data agreed that Narcissistic Numbers were interesting, but not as good as his favorite: Perfect Numbers. Geordi had never heard of a Perfect Number, so Data elaborated, "A positive integer is said to be Perfect if it is equal to the sum of its positive divisors less than itself. For example, 6 is Perfect because $6 = 1 + 2 + 3$."

Geordi began thinking about an algorithm to determine if a number was Perfect, but did not have the raw computing ability of Data. He needs a program to determine if a given number is Perfect.

Help Geordi write that program.

Input

Input consists of a single entry per line. Each line contains a single positive integer n, where $2 < n < 100,000$ for each case. A line containing -1 denotes the end of input and should not be processed.

Output

For each case, determine whether or not the number is Perfect. If the number is Perfect, display the sum of its positive divisors less than itself. The ordering of the terms of the sum must be in ascending order. If a number is not Perfect, print "`<NUM> is NOT perfect.`" where `<NUM>` is the number in question. There must be a single space between any words, symbols, or numbers in all output, with the exception of the period at the end of the sentence when a number is not perfect.

Sample Input

```
6
12
28
-1
```

Sample Output

```
6 = 1 + 2 + 3
12 is NOT perfect.
28 = 1 + 2 + 4 + 7 + 14
```

Problem G — limit 5 seconds

Generations of Tribbles

Tribbles are the cute, fuzzy, cuddly animals that have voracious appetites and reproduction rates that rival any complex organism in the galaxy (tribbles are born pregnant!). After being introduced to the Enterprise and its crew, it was quickly discovered what a nuisance tribbles could be. In a very short amount of time, tribbles were everywhere on the ship.

Fortunately for the Enterprise, Engineer Scott was able to transport them to a nearby Klingon vessel. The Klingons were unaware of the issues tribbles could cause and brought them into Klingon space, where the tribbles spread like locusts and devastated ecosystems of planets across the Klingon Empire.

Members of the United Federations of Planets (The Federation) found this extremely amusing and used the calculation of tribble reproduction as an academic exercise for first year students at its academy.

The following sequence of numbers represents how tribbles reproduce. The first number represents generation 0, the second generation 1, and so on.

$$1, 1, 2, 4, 8, 15, 29, 56$$

The following recurrence can be used to represent the above sequence, where n represents the generation number:

$$\begin{array}{rc} n < 2: & 1 \\ n = 2: & 2 \\ n = 3: & 4 \\ n > 3: & \mathrm{gen}(n-1) + \mathrm{gen}(n-2) + \mathrm{gen}(n-3) + \mathrm{gen}(n-4) \end{array}$$

Those at the academy that know something about old Earth history have jokingly called the recurrence 'Tribblenacci'.

Your job as a first year student at the academy is to accurately and rapidly calculate how many tribbles there will be for a given 'Tribblenacci' number. The fact is, evaluating the above recurrence recursively is slower than chemical propulsion for interstellar travel! To do so for more than a handful of generations would clearly be illogical.

Input

The first line of input will be an integer t $(0 < t < 69)$ representing the number of test cases. Following this will be t integer values, one per line. Each of these will represent a generation number g $(0 \leq g \leq 67)$ to calculate.

Output

For each generation number read, display the corresponding 'Tribblenacci' value.

Sample Input	Sample Output

```
8
0
1
2
3
4
5
30
67
```
	```
	1
	1
	2
	4
	8
	15
	201061985
	7057305768232953720
	```

Problem H — limit 5 seconds

Holodeck Hacking

Someone put a physical mirror in the holodeck chamber, and it has scrambled some critical simulation data! The mirror had the effect of superimposing reversed data on top of the original data. You need to figure out how scrambled the data is.

Given a positive number Y, calculate the number of distinct positive values of X such that $Y = X + \mathrm{rev}(X)$. The rev operator reverses the digits of a number. The values for X must be in their normal decimal form, without leading zeros. For example, $\mathrm{rev}(350) = 53$ and $\mathrm{rev}(53) = 35$.

Input

Input begins with a line with one integer T ($1 \leq T \leq 500$) denoting the number of test cases. Each test case consists of a single line with a single integer Y ($1 \leq Y < 10^{18}$).

Output

For each test case, print out a line containing the count of positive integers that, when summed with their reverse, equals the input value.

Sample Input	**Sample Output**
`8`	`1`
`10`	`1`
`11`	`9`
`121`	`1`
`299999999999999981`	`0`
`109`	`0`
`7087`	`0`
`59284`	`1`
`10201`	

PROBLEM I — LIMIT 5 SECONDS

Interstellar Trade

As a rare reward for passing one of his inscrutable tests, Q has offered Commander Sisko the opportunity to relocate both endpoints of the Bajoran wormhole (and Deep Space Nine along with it, of course). The Commander has asked for proposals for the relocation, and various merchants wish to relocate the endpoints into known space to decrease the transit time between planets known for their commerce. Your job is to figure out how to place the endpoints of the wormhole so as to minimize the maximum distance between any pair of these planets.

Conveniently, all of the planets of interest lie on a straight line, and in the absence of the wormhole, the distance between any pair of them is simply the straight-line distance. Once the wormhole has been added, a traveler has the additional option of going from one planet straight to one end of the wormhole, and then straight from the other end of the wormhole to the other planet. The distance traveled in this case is then the sum of those two distances (travel between the two ends of the wormhole is instantaneous). Note that a traveler always has the option of not using the wormhole, even if an endpoint of the wormhole lies directly between the two planets of interest. Finally, you may place a wormhole endpoint arbitrarily close to any planet, such that the distance from the planet to the wormhole is effectively zero.

Input

Input begins with a line with one integer T ($1 \leq T \leq 50$) denoting the number of test cases. Each test case begins with a line with a single integer N ($2 \leq N \leq 4000$) denoting the number of planets. This is followed by N lines with a single integer x_i each ($-10^9 \leq x_i \leq 10^9$) denoting the location of planet i (all planets are points on the x-axis). No two planets will be at the same location.

Output

For each test case, print on a single line the maximum distance between any pair of planets after the wormhole has been placed in such a manner as to minimize this value. If this distance is fractional, round it up to the next integer. Note that although all planet locations are given as integers, the wormhole location may not have integer coordinates.

Sample Input	Sample Output

```
3
3
-1
1
10
2
1000000000
-1000000000
5
1
2
6
7
8
```
	```
	2
	0
	2
	```

Problem J — limit 75 seconds

Janeway's Journey

Captain Janeway wants to take the Voyager through an asteroid field, but there are too many asteroids for such a trip to be safe. To help blaze a path, you have been asked to take a shuttle into the asteroid field. Your plan is to fly through the field and note its layout, then position the shuttle in such a manner to take out as many asteroids as possible with a single, straight-line phaser blast

For simplicity we will model this problem in the plane, with circular asteroids, an infinitely thin phaser, and the ability to position your shuttle wherever and however you like, in or around the asteroid field.

Input

Input begins with a line with one integer T ($1 \leq T \leq 25$) denoting the number of test cases. Each test case begins with a line with a single integer N denoting the number of asteroids; $1 \leq N \leq 2000$. Furthermore, there will be at most 5 test cases for which $N > 500$. This line is followed by N lines each with 3 space-separated real numbers x_i, y_i, r_i, which specify that asteroid i is centered at (x_i, y_i) and has radius r_i. $-10^6 \leq x_i$, $y_i \leq 10^6$, $1 \leq r_i \leq 100$. All of these numbers will be given to two decimal places. No asteroids will overlap or intersect each other. The input will be such that the answer will not change even if the radii of the asteroids vary by 10^{-6} in either direction.

Output

For each test case, print a single integer on its own line denoting the maximum number of asteroids that may be destroyed.

Sample Input	Sample Output

```
1
3
0.00 0.00 1.00
3.00 0.00 1.00
3.00 3.00 1.00
```
	`2`

Problem K — limit 10 seconds

Klingon Warfare

War has broken out between two Klingon clans. But war between Klingons is not just any war; it must be both honorable and glorious. For honor, the two sides must be exactly matched. For glory, there must be as many participants as possible.

Each clan obeys a strict hierarchy: there is one leader of the entire clan. This leader may have zero or more direct subordinates who are ordered from eldest to youngest. Each subordinate in turn may have zero or more direct subordinates of his or her own, also ordered from eldest to youngest (and so on and so forth). By tradition, every clan warrior is younger than his or her superior. Furthermore, each individual clan warrior specializes in one of several distinct fighting styles.

The subclan commanded by a clan warrior consists of the warrior himself and all direct or indirect reports (i.e., subordinates, subordinates of subordinates, etc.). A pair of subclans are said to match exactly if two conditions are met. First, the leaders of each subclan must have the same fighting style and number of subordinates. Second, assuming that the direct subordinates of each leader are ordered by decreasing age, then the subclans commanded by the first direct subordinates must match exactly, the subclans commanded by the second direct subordinates must match exactly, and so on.

Each clan will select its participants in the war, consisting of a single warrior and his or her subclan. The two subclans chosen must match exactly, and must be as large as possible. How many warriors fight for each clan?

Input

Input begins with a line with one integer T ($1 \leq T \leq 50$) denoting the number of test cases. Each test case begins with a line with two integers M and N ($1 \leq M$, $N \leq 10000$) denoting the size of the two clans. Next follow M lines with an uppercase letter f_i and an integer s_i ($s_0 = -1$, $0 \leq s_i < i$, zero-indexed) denoting the fighting style and the superior respectively of warrior i of the first clan. Warrior 0 is always the clan leader (and has a "superior" of -1 to indicate this). Warriors are ordered from eldest to youngest. Next follow N lines with an uppercase letter f_j and an integer s_j (same bounds) denoting the fighting style and the superior respectively of warrior j of the second clan.

Output

For each test case, print out a line with a single integer equal to the maximum number of warriors that each clan may send to fight.

Sample Input

```
1
3 4
A -1
B 0
C 0
Z -1
A 0
B 1
C 1
```

Sample Output

```
3
```

Problem L — limit 5 seconds

Languages

The Enterprise has encountered a planet that at one point had been inhabited. The only remnant from the prior civilization is a set of texts that was found. Using a small set of keywords found in various different languages, the Enterprise team is trying to determine what type of beings inhabited the planet.

Input

The first line of input will be N ($1 \leq N \leq 100$), the number of different known languages. The next N lines contain, in order, the name of the language, followed by one or more words in that language, separated with spaces. Following that will be a blank line. After that will be a series of lines, each in one language, for which you are to determine the appropriate language.

Words consist of uninterrupted strings of upper or lowercase ASCII letters, apostrophes, or hyphens, as do the names of languages. No words will appear in more than one language.

No line will be longer than 256 characters. There will be at most 1000 lines of sample text.

Every sample text will contain at least one keyword from one of the languages. No sample text will contain keywords from multiple languages. The sample text may contain additional punctuation (commas, periods, exclamation points, semicolons, question marks, and parentheses) and spaces, all of which serve as delimiters separating keywords. Sample text may contain words that are not keywords for any specific language.

Keywords should be matched in a case-insensitive manner.

Output

For each line of sample text that follows the blank line separating the defined languages, print a single line that identifies the language with which the sample text is associated.

Sample Input

```
4
Vulcan throks kilko-srashiv k'etwel
Romulan Tehca uckwazta Uhn Neemasta
Menk e'satta prah ra'sata
Russian sluchilos

Dif-tor heh, Spohkh. I'tah trai k'etwel
Uhn kan'aganna! Tehca zuhn ruga'noktan!
```

Sample Output

```
Vulcan
Romulan
```

Problem M — limit 5 seconds

Mass Production

Star Fleet needs to deploy a squadron of ships at the edge of Federation space. There is a nearby planet where the ships can be built, but the planet doesn't have the infrastructure to support the construction of ships from scratch. It is, however, possible to assemble the ships using prefabricated kits containing an assortment of base parts. Each kit is designed to be turned into a ship by converting the base parts into the necessary ship components. To ensure consistent construction, parts from different kits should not be mixed and matched; each kit must be used in its entirety to construct exactly one ship. This squadron consists of two distinct classes of ships, Class A ships and Class B ships. Both classes consist of the same total number of components, though their individual makeup is different. Each base part is capable of being turned into any type of ship component, though the cost to turn a base part into a ship component varies depending on the base part type and the ship component type. You are responsible for creating these prefabricated kits, which must all be identical to each other. You have access to M5, the greatest computer of all time. What should the composition of the kit be to minimize the total cost of constructing the squadron?

Input

Input begins with a line with one integer T ($1 \leq T \leq 50$) denoting the number of test cases. Each test case begins with a line with four integers M, N, A, and B ($1 \leq M, N \leq 10$; $1 \leq A, B \leq 100$), where M denotes the number of types of base parts, N denotes the number of types of ship components, A denotes the number of Class A ships required, and B denotes the number of Class B ships required. Next is a line with N integers a_i denoting the quantity of ship component i needed for each Class A ship ($0 \leq a_i \leq 100$). Next is a line with N integers b_i denoting the quantity of ship component i needed for each Class B ship ($0 \leq b_i \leq 100$; $\sum_i a_i = \sum_i b_i$). Next follow M lines with N integers c_{ij} ($0 \leq c_{ij} \leq 100$) denoting the cost of converting a single base part i into a single ship component j.

Output

For each test case, output a line with a single integer equal to the minimum total conversion cost of assembling all the ships from the factory kits.

Note

In the sample given, the optimal factory kit has one of each base part; such a kit costs 1 to convert into the components for a Class A ship and 2 to convert into the components for a Class B ship.

Sample Input

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

Sample Output

```
14
```