



Problems

- A. A Sourvenir
- B. Milk Can
- C. Beauty Prime Numbers
- D. Divisible by 7
- E. Prof. PVH and the country of 16
- F. Palindrome Counter
- G. All for The Central Region
- H. A Kind of Multicore Virtual Server
- I. Crystal Ball
- J. Evacuation
- K. Assigning course
- L. Constant strings
- M. Supsub

Do not open before the contest has started.

Advice, hints, and general information

- The problems are not sorted by difficulty.
- Your solution programs must read input from *standard input* (e.g. System.in in Java or cin in C++) and write output to *standard output* (e.g. System.out in Java or cout in C++). For further details and examples, please refer to your administrator guide and Domjudge documentation.
- For information about which compiler flags and versions are used, please refer to your administrator guide. (Python 2.7.17, Oracle Java 1.8.0_144, gcc 7.5.0 (C, C++ std14)).
- Your submissions will be run multiple times, on several different inputs. If your submission is incorrect, the error message you get will be the error exhibited on the first input on which you failed.
 - E.g., if your instance is prone to crash but also incorrect, your submission may be judged as either “Wrong Answer” or “Run Time Error”, depending on which is discovered first. The inputs for a problem will always be tested in the same order.
- If you think some problem is ambiguous or underspecified, you may ask the judges for a clarification request through the Domjudge system. The most likely response is “No comment, read problem statement”, indicating that the answer can be deduced by carefully reading the problem statement or by checking the sample test cases given in the problem, or that the answer to the question is simply irrelevant to solving the problem.
- In general, we are lenient with small formatting errors in the output, in particular whitespace errors within reason, and upper/lower case errors are often (but not always) ignored. But not printing any spaces at all (e.g. missing the space in the string “1 2” so that it becomes “12”) is typically not accepted. The safest way to get accepted is to follow the output format exactly.
- For problems with floating point output, we only require that your output is correct up to some error tolerance. For example, if the problem requires the output to be within either absolute or relative error of 10^{-4} , this means that
 - If the correct answer is 0.05, any answer between 0.0499 and .0501 will be accepted.
 - If the correct answer is 500, any answer between 499.95 and 500.05 will be accepted.
- Any reasonable format for floating point numbers is acceptable. For instance, “17.000000”, “0.17e2”, and “17” are all acceptable ways of formatting the number 17. For the definition of reasonable, please use your common sense.

Problem A

A Sourvenir

Time Limit: 1 seconds

Memory Limit: 512 Megabytes

Problem description

Mr. Bean is selected to be a contestant member of Elon Musk's University which is located in Mars to compete in ICPC Can Tho. This will be the first time he visits the city of Can Tho and the Mekong Delta.

Next to the hotel he stays in Can Tho is a souvenir shop where you can buy all specialties of this region. The shop has N items, numbered 1 to N . Item i has the price of i dong. Bean has K dong and he wants to buy as many distinct items as possible.

Given N and K , your task is to calculate how many different items Mr. Bean can buy?

Input

The input consists of multiple test cases. It starts with T - the number of test cases. Then each test case is printed in one line, containing two integers N and K . ($1 \leq N \leq 100$, $0 \leq K \leq 10000$).

Output

For each test case, print the maximal number of souvenirs Mr. Bean can buy.

Example:

Input
2
10 4
10 1000
Output
2
10

A relax page, open to next page for the next challenge in your journey to the TOP.

Problem B

Milk Can

Time Limit: 1 seconds

Memory Limit: 256 Megabytes

Problem description

A big food company produces canned milk. The can is a round cylinder. They produce various sizes of milk cans. When producing milk cans, the designers of this company always aim to have the lowest cost of raw materials for making cans. That means the total surface area of the can is the smallest. But they expect the volume of the milk can to remain constant, equal to V ml.

You are the talented designers here. Calculate the optimal total surface area of the can for your company to spend the least money.

Notes:

- The volume of cylinder = $\pi * r^2 * h$ cubic units.
- Surface area of cylinder = $2 * \pi * r^2 + 2 * \pi * r * h$.

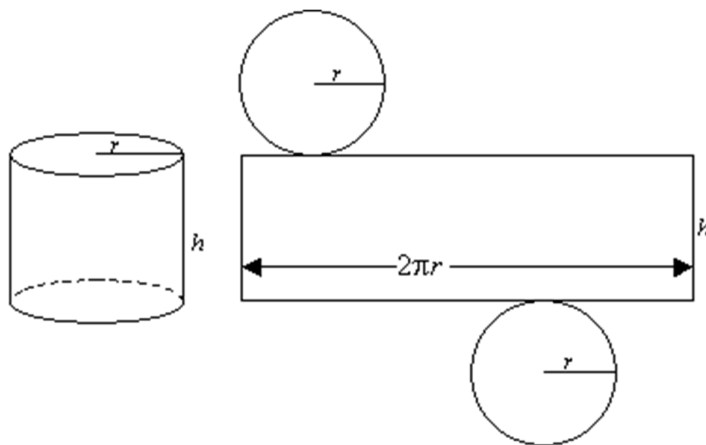


Figure 1. A milk can size specification

Input

The input consists of several test cases. The first line of the input contains the number of test cases T ($T \leq 1000$). The following lines describe the test cases.

Each test case consists of a positive integer V ($V \leq 1000$).

Output

For each test case in the input, print in a single line the smallest surface area with absolute or relative error less than 10^{-6} that you can achieve.

Example:

Input
3
325
500
1000
Output
261.679751
348.734205
553.581045

Problem C

Beauty Prime Numbers

Time Limit: 1 seconds
Memory Limit: 256 Megabytes

Problem description

The significance of prime numbers, in both everyday applications & as a subtopic pertinent to all branches of math, cannot be overstated. We quietly rely on their special properties to carry the backbone of countless parts of our society — all because they are an irreducible part of the very fabric of nature. Resistant to any further factorization, prime numbers are often referred to as the “atoms” of the applied information technology.



You are the talented programmer, and be asked to write a program to count all beauty prime numbers. A beautiful prime number is defined as a number which sum of all digits in it modulo 10 equals to zero (0).

Input

The input consists two integer numbers (m, n). The first number (m) is the start number which we start to find out these beauty prime numbers. The second number (n) is the amount of prime numbers start from first number, in which we want to find out these beauty prime numbers.

$$0 < m, n \leq 10000$$

Output

For each test case in the input, print out the number of beauty prime numbers which program find out. In case of don't have any beauty prime number, print out zero value (0).

Example 1:

Input
8 5
Output
1

Explanation: start from 8, we find out 5 prime numbers: 11, 13, 17, 19, 23. And only one number 19 has $(1+9) \bmod 10 = 0$, so output is 1.

Example 2:

Input
101 10
Output
2

Explanation: start from 101, we find out 10 prime numbers: 101 103 107 109 113 127 131 137 139 149. And there are two numbers 109 ($1+0+9$) and 127 ($1+2+7$) have $\text{mod } 10 = 0$, so output is 2

Problem D

Divisible by 7

Time Limit: 2 seconds
Memory Limit: 512 Megabytes

Problem description

Your task is to write a program that takes an integer as input, reverses its digits, creates the sum of the original integer with its reversed counterpart, and checks if the sum is divisible by 7. If it is, you are done. Otherwise, we repeat the procedure until it does.

If it took more than 1,000 iterations, we assume that there is no multiples 7 can be created by using this method with the input number.

Input

The first line of the contains the number of datasets, which is not greater than 1000.

Each test case contains a positive number N ($1 \leq N \leq 10^{18}$).

Output

For each test, print the final sum that you have found. If there is no answer, print -1.

Example:

Input
5
1
2
3
4
999999
Output
77
77
9447438
77
999999

Explanation for the first test case: $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \rightarrow 16 \rightarrow 77$.

A relax page, open to next page for the next challenge in your journey to the TOP.

Problem E

Prof. PVH and the country of 16

Time Limit: 2 seconds
Memory Limit: 512 Megabytes

Problem description

Prof. PVH is the president of the country of 16. The country has n cities connected by m **directed** roads.

A graph is said to be strongly connected if every vertex is reachable from every other vertex. The strongly connected components of a directed graph form a partition into subgraphs that are themselves strongly connected.

During this COVID-19 pandemic, our Prof is bored, so he has come up with a fabulous idea of reversing the direction one road. He needs to find out, for the i^{th} road $u_i \rightarrow v_i$, if he reverses the direction of this road to $v_i \rightarrow u_i$ and keeps the direction of the other $m-1$ roads, will it change the number of strongly connected components.

Our Prof. has promised to treat you with delicious cups of bubble tea if you can help him solve this boring problem. Let's get it!

Input

The first line contains two integers n, m ($1 \leq n \leq 1000, 1 \leq m \leq 2 * 10^5$) – the number of cities and the number of roads.

Next m lines, each contain two integers u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$) represents a directed edge from u_i to v_i

It's guaranteed that between any pair of cities there is at most one directed road connecting them.

Output

The output consists of m lines. In the i^{th} line, print Y if reversing the i^{th} road will change the number of strongly connected components, and print N otherwise.

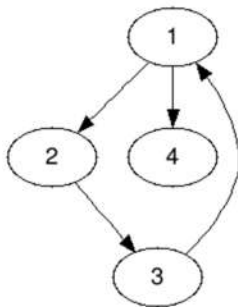
Example 1:

Input	Output
4 4	Y
1 2	Y
2 3	Y
3 1	N
1 4	

Example 2:

Input	Output
5 9	Y
3 4	N
1 3	N
5 2	N
5 3	N
1 2	Y
4 1	N
1 4	Y
2 3	Y
1 5	

Explanation:



For the 1st example, the original number of SCC is 2. If we reverse one of the three edge $1 \rightarrow 2$, $2 \rightarrow 3$ and $3 \rightarrow 1$, the number of SCC will be 4, hence we print *Y* for those three edges. If we reverse edge $1 \rightarrow 4$, the number of SCC will still be 2, hence we print *N* for it.

Problem F

Palindrome Counter

Time Limit: 1 seconds
Memory Limit: 512 Megabytes

Problem description

A palindromic number is a number (in some base b) that is the same when written forwards or backwards. The first few palindromic numbers are therefore are 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101, 111, 121, ... (**rule 1**).

In some special cases, 0012100, 00 is a palindromic number. Therefore, 12100, 1210, 00 is also a palindromic number (**rule 2**).

We can classify positive integers into three categories:

- Type 0: numbers that not palindromic number.
- Type 1: palindromic numbers according to rule 1.
- Type 2: palindromic numbers according to rule 2.

All palindrome that less than or equal to 100 are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 20, 22, 30, 33, 40, 44, 50, 55, 60, 66, 70, 77, 80, 88, 90, 99, 100.

Given a number M is a non-negative integer. The function $F(M)$ returns the number of palindromes (including the 2 types: type 1 and type 2) less than or equal to M ($0 \leq M \leq 10^{35}$).

M	0	15	39	55	60	85	88	90	100
F(M)	1	12	16	20	21	25	26	27	29

To simplify the problem, given a number N is a non-negative integer, the function $P(N)$ returns the value of $F(10^N)$ ($0 \leq N \leq 35$).

N	0	1	2	3	4	...	15	...	35
P(N)	2	11	29	137	335	...	144,444,413	...	1,444,444,444,444,444,373

Task

Given a N positive numbers, your mission is to calculate $P(N)$.

Input

- On the first line is the number of test case, $1 \leq T \leq 36$.
- In the next T lines, each line contains a non-negative integer N_i ($1 \leq i \leq T$, $0 \leq N_i \leq 35$).

Output

The result has T lines, the i^{th} line contains the value of $P(N_i)$ ($1 \leq i \leq T$).

Example 1:

Input	Output
5	335
4	14429
7	1444444444444444373
35	34444444403
20	144444413
15	

Example 2:

Input	Output
3	11
1	34427
8	344423
10	

Example 3:

Input	Output
5	344444444444444383
30	29
2	137
3	144444444444444377
33	14444444444393
25	

Problem G

All for The Central Region

Time Limit: 2 seconds

Memory Limit: 512 Megabytes

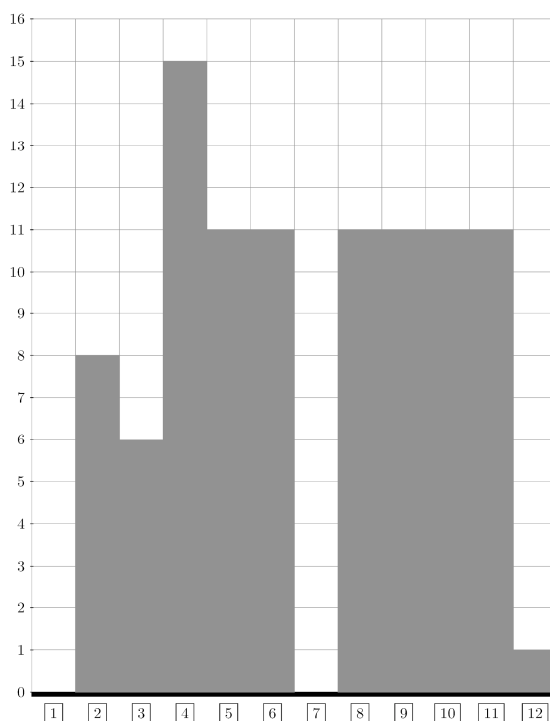
Problem description

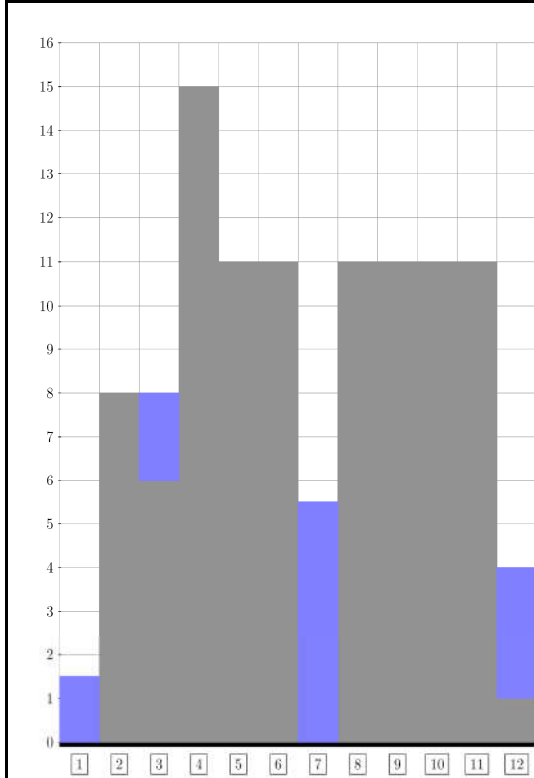
This year's stormy situation is going very complicatedly in the Central region, causing very heavy damage. These days, heavy rains appear again, causing widespread flooding.

In a valley surrounded by high mountains, the water inside cannot flow out and vice versa. The topography in the area is represented on the Ox coordinate axis consisting of N areas with the same size, the array A contains N integers which are the altitudes of them. In the near future, it will rain with precipitation H , meaning that per unit of rain will cause water levels of elevation H (in theory). Rainwater will flow from high to low. If a group is **higher** than its 2 neighbors, the water will flow to the side with equal volume.

To be precise, let $A[i]$, $F[i]$ be the altitude and the water amount of the i -th area. We can see that the surface of the i -th area will be at the altitude of $A[i] + F[i]$. Adjacent areas with the same water surface form a group. If a group's surface altitude is higher than its 2 neighbouring groups, the additional rain water will flow to its neighbors (the amount is divided evenly) until we are clear of rain or the water level of its neighbor rises and they form a bigger group.

Consider the following example:





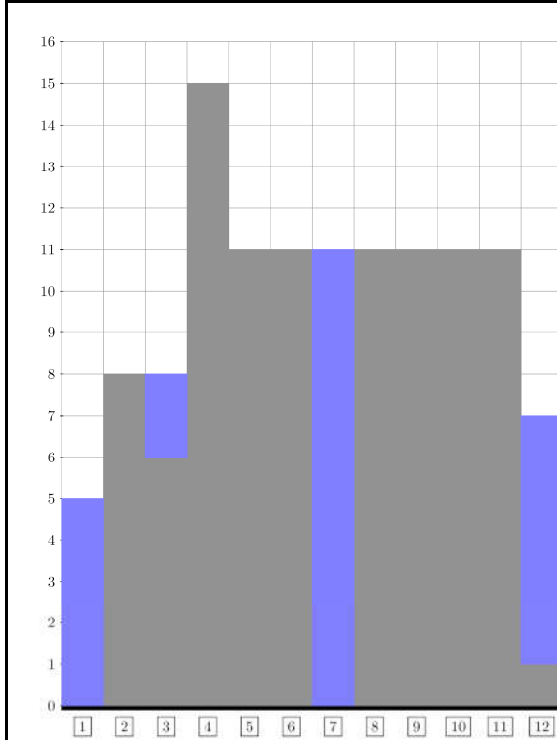
This is $H=1$. Note that:

Area 1 is getting 1 from the rain in place, 0.5 from area 2, thus $F[1] = 1.5$

Area 3 is getting 1 from the rain in place, 0.5 from area 2 and 0.5 from area 4, thus $F[3] = 2$.

Area 7 is getting 1 from the rain in place, 0.5 from area 2, all 2 units from area 5 and area 6 and half of the group contains area 8, 9, 10, 11. Thus $F[7] = 5.5$.

Area 12 is getting 1 from the rain in place and half of the group contains area 8, 9, 10, 11... Thus, $F[12]=3$

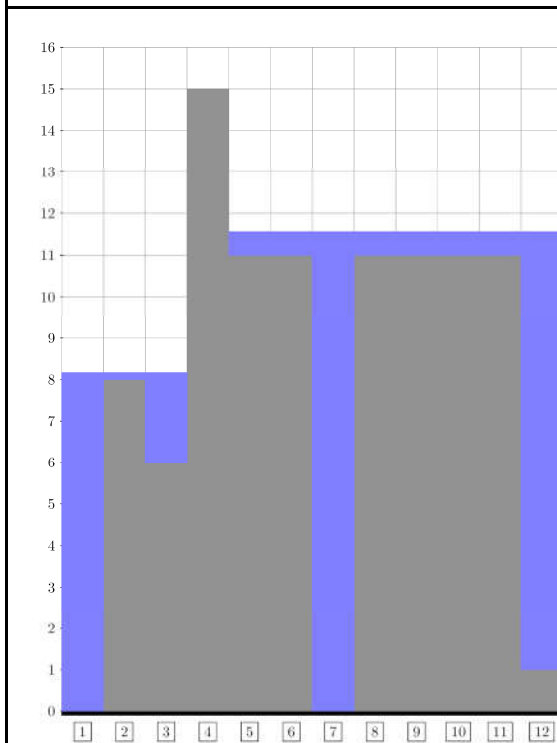


This is $H=2$. Note that area 2 and area 3 are now forming a group.

Area 1 will get 1 from the rain in place, 0.5 from area 4 and all 2 units from area 5 and 6. Thus, $F[1] = 5$

Area 7 will get extra 5.5 like the last time.

Area 12 will get extra 3 like the last time.



This is $H=3$. Note when $H=2$, area 7 to 11 are forming a new group.

Given the area's topography and precipitation, determine the water level in each area.

Input

The first line includes T ($1 \leq T \leq 10$), the number of test cases.

The first line of each test case consists of 2 integers N and H . Number of areas in the valley and precipitation ($1 \leq N \leq 1000, 1 \leq H \leq 10^5$).

The next line contains N integers A_i ($1 \leq i \leq N, 0 \leq A_i \leq 10^5$)- Altitude of each area.

Output

Including T lines corresponding to each test case.

Each line contains N decimals accurate to 10^{-4} as the result of each test case.

Example:

Input	Output
3	1.5 1.5 0
3 1	1.5 0 2 0 0 0 5.5 0 0 0 0 3
1 1 3	5 0 2 0 0 0 1 1 0 0 0 0 6
12 1	
0 8 6 15 11 11 0 11 11 11 11 1	
12 2	
0 8 6 15 11 11 0 11 11 11 11 1	

Problem H

A Kind of Multicore Virtual Server

Time Limit: 3 seconds
Memory Limit: 512 Megabytes

Problem description

Nam, Bac, Dong, and Tay are students of the computer science department at ACM Lab, and they have each one a classical computer with one core. Recently, they studied multi-cores CPU, and they would like to build up a simulation multi-cores CPU Server by creating a software run on a computer owned by the department to combine their own 4 classical computers into a virtual one.

The simulation software will receive tasks raised by the department computer's operating system in each time-sliced unit of computation. Some tasks reached simulation software at the same time. Each task will be assigned a PID for identification. When a task reached the simulation software in a time-sliced unit, it will be processed in the next time-sliced unit in the simulation software. The simulation software will go around to assigned tasks to CPUs based on its PID when it reached to assigning phase, and the assignment process will start from the CPU, which is the next one with the last assigned CPU in turning around, i.e.

If the last assigned CPU is 1st, the start one is 2nd CPU.

If the last assigned CPU is 2nd, the start one is the 3rd CPU.

If the last assigned CPU is 3rd, the start one is 4th CPU.

If the last assigned CPU is 4th, the next start is 1st.

And so on.

In each CPU, a task's processing time consumption – called burst time - will also be sliced into some unit pieces, and it will be allocated into the CPU computation time slots in around.

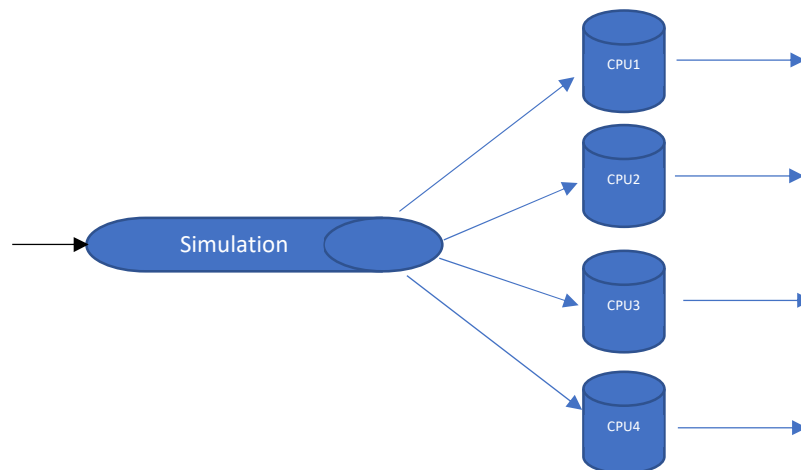


Figure 2. Simulation and CPUs structure

When a task reaches a CPU, it will be allocated in processing waiting list, and the CPU will assign them a fixed time slot in a cyclic way for processing. In case a Task/Process had been completed earlier than the end of time slot, the CPU will immediately pick the next Task/Process in the processing wait list to execute.

1. Completion time: Time at which process completes its execution.
2. Turn Around Time: Time Difference between completion time and CPU arrival time. $\text{Turn Around Time} = \text{Completion Time} - \text{CPU Arrival Time}$
3. Waiting Time: Time Difference between turn around time and burst time. $\text{Waiting Time} = \text{Turn Around Time} - \text{Burst Time}$
4. Total Turn Around Time: is computed by sum all of the turn around time of processes.
5. Total Waiting Time is the sum of the waiting time of all processes.
6. Average Turn Around Time in each CPU is equal to the total average turn around time divide with the number of tasks/processes.
7. Average Waiting Time in each CPU is equal to the total waiting time divide with the number of tasks/processes.
8. Average Turn Around Time of a task in the Virtual Server is equal to the sum of all Average Turn Around Time in each CPU divide with number of CPUs
9. Average Waiting Time of a task in the Virtual Server is equal to the sum of all Average Waiting Time in each CPU divide with number of CPUs

Assume that the timer in all CPUs and simulation machine are synchronized. The simulation software doesn't care about busy or leisure state of CPUs while assigning processes to them.

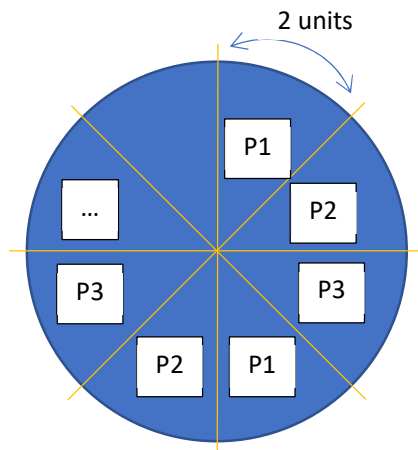


Figure 3. a time slot in CPU processing

Input

First line:

N: number of Tasks/Processes will be executed – integer number ($1 < N \leq 10^3$)

slot: length of a CPU processing slot in time sliced unit – integer number. ($1 < \text{slot} \leq 10^2$)

Next N lines contain the information of each task. i^{th} line contains 3 values, separated by a space

- pid – tasks/processes id - integer number ($0 < \text{pid} < 10^6$)
- bt_i - burst time of task i^{th} , integer number ($0 < \text{bt}_i < 10^6$) - in time sliced unit
- rt_i - reached time of task i^{th} on simulation software, integer number ($-1 \leq \text{rt}_i < 10^6$) - in time sliced unit. In case rt_i is equal -1, the process i^{th} will be assigned to a CPU immediately at the first assigning phase of simulation software.

Output

One lines with 2 values in time sliced unit, separated by a space.

Wt: a floating number - Average waiting time of a task in the Virtual Server

Tt: a floating number - Average turn around time of a task in the Virtual Server

Floating point value accuracy is 10^{-6} .

Example 1:

Input	Output
1 2	0.250000 1.250000
1 4 5	

Example 2:

Input	Output
2 2	0.500000 2.000000
1 4 8	
2 2 9	

Example 3:

Input	Output
2 2	0.500000 2.500000
1 4 0	
2 4 0	

Example 4:

Input	Output
2 2 1 4 -1 2 8 -1	0.500000 3.500000

Problem I

Crystal Ball

Time Limit: 1 seconds
Memory Limit: 512 Megabytes

Problem description

Tuan is a famous prophet because of his predictability, as he also has a special super-power. He owns a crystal ball with radius $r = 1$. When the ball is positioned so that its center is as the origin $O(0,0,0)$ in 3-dimensional space, it can float in the air and rotate around its center in any direction. When predicting a certain event, Tuan focuses on 3 points $A(x_A, y_A, z_A)$, $B(x_B, y_B, z_B)$ and $C(x_C, y_C, z_C)$ on the surface of the sphere. With his special super-power, the ball will automatically rotate around \overrightarrow{OP} with an angle equal to the angle between 2 vectors \overrightarrow{OB} and \overrightarrow{OC} , where \overrightarrow{OP} is a unit normal vector of OBC plane. At that time, the coordinates of points A, B, and C will be changed. Depending on the changes, Tuan will predict future events.



Your task is to find out the new coordinate of point A after changed.

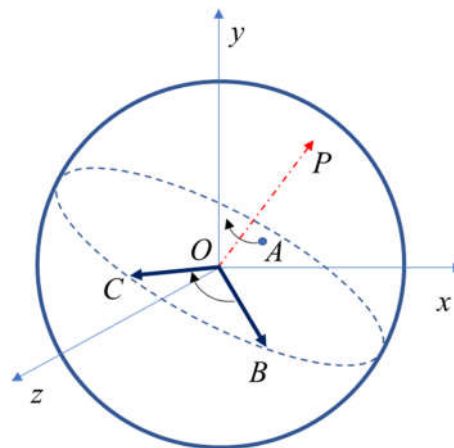


Figure 4. A Sphere

Input

The first line includes T ($1 \leq T \leq 100$), the number of test cases.

The first line of each test case consists of 6 real numbers $x_A, y_A, x_B, y_B, x_C, y_C$, where $x_A^2 + y_A^2 \leq 1$, $x_B^2 + y_B^2 \leq 1$ and $x_C^2 + y_C^2 \leq 1$.

The second line contains 3 integers s_A, s_B and s_C ($0 \leq s_A, s_B, s_C \leq 1$). $s_A = 1$ means that $z_A \geq 0$ and otherwise $z_A < 0$; and similarity for points B and C .

Output

Including T lines corresponding to each test case.

Each line contains 3 decimals as the new coordinate of point A .

Your answer will be accepted if its relative or absolute error does not exceed 10^{-6} .

Example:

Input	Output
3	0.86602540 0.5 0.0
0.0 0.5 0.0 0.0 1.0 0.0	0.0 0.0 1.0
1 1 1	0.0 1.0 0.0
0.5 0.5 0.5 0.5 0.0 0.0	
0 0 1	
0.5 0.5 0.5 -0.5 -0.5 0.5	
1 1 1	

Problem J

Evacuation

Time Limit: 2 seconds

Memory Limit: 512 Megabytes

Problem description

In a dark room, at approximately 11:55PM, Captain couldn't sleep and begins thinking about possible consequences of a nuclear catastrophe in his city. As its mayor, he is acquainted with the following facts. First, he knows that there are exact N people living in the city, each of them is in their own houses. Second, there are M roads connecting the houses. For each road, he knows how much time to travel from one house to the other house that it connects. Finally, Captain knows that there are K houses with nuclear bunkers. Each of these houses has one bunker that can accommodate up to a certain number of people. With all these in his mind, Captain makes the following question: "How much time does it take to evacuate all the residents of the city?"

Please help Captain answer the question given the following assumptions:

- (i) the residents can move optimally (they know which way is the shortest),
- (ii) multiple residents can move on a same road simultaneously in different directions,
- (iii) there is at least one path connecting any two houses.

Note that, after the evacuation, every resident ends up in one of the nuclear bunkers.

Input

- The first line contains the integers N , M , and K ($1 \leq N \leq 10^5$, $1 \leq M \leq 3 \times 10^5$, $1 \leq K \leq 17$) which respectively represent the number of residents, the number of roads, and the number of nuclear bunkers. The houses are made with the numbers from 1 to N .
- In each of the following M lines, there are three integers A , B , and C ($1 \leq A, B \leq N$, $A \neq B$, $1 \leq C \leq 10^9$), which means that the time to travel from the house A to the house B is C (units of time).
- Each of the next K lines contains two integers X and Y ($1 \leq X \leq N$, $1 \leq Y \leq 10^9$) which indicate that the bunker of the house X can accommodate up to Y people. The total capacity of all the bunkers is greater than or equal to N .

Output

- A single line that prints out the minimum units of time required to evacuate all the residents of the city.

Example:

Input	Output
7 8 3 1 2 5 2 3 3 3 4 5 1 4 1 4 5 7 5 6 2 6 7 1 4 7 4 3 3 7 3 6 2	5

Problem K

Assigning course

Time Limit: 2 seconds
Memory Limit: 512 Megabytes

Problem description

Assume that you are a staff of the university who is in charge of arranging lessons. Suppose you have N time slots, M professors and K lessons. A lesson or a time slot should be assigned to exactly one professor and each professor agrees to teach only one lesson in this semester.

For each professor, he will prefer some certain time slots among these N time slots, and prefer to teach some certain lessons among these K lessons. A professor will be satisfied iff you arrange him both his preferred lesson and preferred time slot. Your goal is to satisfy as many professors as you can.

Design an algorithm to determine how many professors can you satisfy at most.

Input

The first line of the input contains the number of datasets, which is not greater than 10.

Each test case begins with four integers N, M, K, L ($N, M, K \leq 100, L \leq 10^6$), denotes the number of time slots, the number of professors, the number of lessons and the number of descriptions.

The next L lines present descriptions with the standard form “A B C”, means that Prof. B prefers to teach course C at time slot A. A is an integer that $1 \leq A \leq 100$, while B and C are strings of length less than 20.

Output

For each test, print the maximum number professors who are satisfied.

Example:

Input	Output
2	2
2 2 2 3	1
1 JAMES A1	
1 JAMES B1	
2 PETER B1	
3 3 1 3	
1 JAMES A1	
2 PETER A1	
3 BOB A1	

A relax page, open to next page for the next challenge in your journey to the TOP.

Problem L

Constant strings

Time Limit: 2 seconds

Memory Limit: 512 Megabytes

Problem description

A string is considered constant if all characters of the string is the same. For example, “a”, “bb” and “ccc” are constant strings, while “abc” is not.

A substring of a string is obtained by removing a (possibly empty) prefix and a (possibly empty) suffix. Hence, the string “abbab” has 6 constant substrings: “a” appears twice, “b” appears three times and “bb” appears once.

Given two integers n and k , find the k -th string with exactly n constant substrings, when all strings are sorted in lexicographic order.

As a remind, string $s = s_1s_2...s_m$ is considered lexicographically smaller than string $t = t_1t_2...t_n$ if and only if either of the following conditions is satisfied:

- $m < n$ and $s_i = t_i$ for every $1 \leq i \leq m$
- there exists an index i such that $1 \leq i \leq m$, $s_i < t_i$ and $s_j = t_j$ for every $1 \leq j < i$

Input

The input contains several (at most 10) test cases. Each test case is presented in one separate line with two integers n and k ($1 \leq n \leq 1e7$, $1 \leq k \leq 3e18$).

The input is terminated by a line containing two zeros, which is not a test case.

Output

For each test case, print the sought string in a single line. It is guaranteed that such string exists.

Example:

Input	Output
3 1	aa
3 2	aba
3 3	abc
3 4	abd
3 5	abe
0 0	

A relax page, open to next page for the next challenge in your journey to the TOP.

Problem M

Supsub

Time Limit: 1 seconds

Memory Limit: 512 Megabytes

Problem description

Given an array $a = a_1, a_2, \dots, a_n$

The cost of the subsegment a_l, a_{l+1}, \dots, a_r is $\sum_{i=l}^r \sum_{j=l}^i a_j$.

Find the maximum cost of a subsegment of a .

Input

- The first line contains an integer: n ($1 \leq n \leq 10^5$)
- The second line contains n integers: $a_1 a_2 \dots a_n$ ($-10^9 \leq a_i \leq 10^9$)

Output

The maximum cost of a subsegment of a

Example:

Input	Output
7 1 3 -4 3 1 -1 6	24

Look back the scoreboard, are you on the TOP alone?