The 2022 ICPC Asia Topi Regional On-site Contest

- Do not open the booklet unless you are explicitly told to do so. You can only read these instructions below.
- Do not create disturbance or move around unnecessarily in the arena.
- If you have any question regarding the problems, send a clarification from the judges using DOMJudge.
- Internet and mobile phones are also not allowed.
- Before submitting a run, make sure that it is executable via command line. For Java, it must be executable via "javac" and for GNU C++ via "g++". Java programmers need to remove any "package" statements and source code's file name must be the same as of main class. C++ programmers need to remove any getch() / system("pause") like statements.
- Do not attach input files while submitting a run, only submit/attach source code files, i.e., *.java or *.cpp or *.py.
- Language supported: C/C++, Java and Python3.
- Source code file name should not contain white space or special characters.
- You must take input from Console, i.e., Standard Input Stream (stdin in C, cin in C++, System.in in Java, stdin in Python)
- You must print your output to Console, i.e., Standard Output Stream (stdout in C, cout in C++, System.out in Java).
- Please, don't create/open any file for input or output.
- Please strictly meet the output format requirements as described in problem statements, because your program will be auto judged by computer. Your output will be compared with judge's output byte-by-byte and not tolerate even a difference of single byte. So, be aware! **Pay special attention to spaces, commas, dots, newlines, decimal places, case sensitivity, etc.**
- **Pay special attention while taking input. Some languages may include carriage return as an extra line.**
- Unless mentioned in some problem, all your programs must meet the time constraint of 5 seconds.
- The decision of judges will be final.
- **The link for DomJudge is:** https://domjudge2.giki.edu.pk/domjudge/public

PROBLEM 1: PCN-Bot

Time limit: 5 seconds

You are designing a mechanism to scrape the internet using multiple web-bots. A web-bot may hop from one website to another website if there is a hyperlink from the first website to the next. We would like Web-bot(s) hopping among all websites such that they hop back to the first one again (after some number of hops) where they started. A web-bot may get stuck on a website if there is no outgoing hyperlink on the website.

A web-bot can be launched from a website at some cost. Some websites have a higher launch cost than the others. It is possible that you might need to launch multiple web-bots from different websites to scrape the complete internet. Given a network of websites and the costs of launching web-bots from each website, calculate the following values:

- Determine the minimum possible cost of launching web-bots to scrape the complete network.
- Determine the minimum number of web-bot(s) required to ensure that no website is missed.
- Determine the number of all possible ways to satisfy the first two constraints. Two ways are considered different if they have different starting websites.
- Determine the largest count of websites that can be scraped using a single web-bot.

Input

In the first line, you will be given an integer n, number of websites ($1 \le n \le 3 \cdot 10^5$). In the next line, n space-separated integers will be given. The i^{th} integer is the cost c_i of launching a web-bot from the i^{th} website (costs will be non-negative and will not exceed 10^9).

The next line will contain an integer m ($0 \le m \le 3 \cdot 10^5$). And each of the next m lines contains two integers u_i and v_i ($0 \le u_i, v_i \le$ n - 1; $u \ne v$). A pair u_i, v_i means, that there is a hyperlink from u_i to v_i. There will not be more than one hyperlink between two nodes in the same direction.

Output

Print four integers separated by spaces. The first one is the minimum possible cost of launching web-bots to scrape the internet. The second one is the number of ways you can launch the web-bots. The third integer represents the largest count of websites that can be scraped using a single web-bot. And the fourth integer represents the minimum number of web-bot launches required to scrape the internet.

Sample input	Sample Output
3 1 2 3 3 0 1 1 2 2 1	3 1 2 2
5 2 8 0 6 0 6 0 3 0 2 1 3 2 3 3 4 4 0	8 2 4 2

PROBLEM 2: Assignment Similarity Check

Time limit: 5 seconds

Some students have submitted their assignment to you as a teacher of the Computer Programming course. Being a suspicious teacher, you want to run a similarity check on these assignments to see how similar they are. For this purpose, you will use a simple string similarity metric called the “Levenshtein Distance (LD)”. The LD is defined between two input strings (string1 & string2) as the minimum number of character insertions, deletions, or substitutions that would be required to convert string1 to string2.

Some examples are given below:

- LD between string “Hello” and “Helli” is 1 (1 for substituting o with i)
- LD between string “Hello_world” and “Hi” is 10 (1 for substituting e with i, and 9 for deleting llo_world)

For a more exhaustive explanation, refer to the LD table for "CHEQUE" and "CHECK" below. The value at cell (i, j) denotes the LD of prefix of "CHECK" of length i and prefix of "CHEQUE" of length j. For e.g: cell (i=1, j=2), the LD is 1, which is the LD between strings "C" and "CH".
Cell (5, 6) denotes LD between "CHECK" and "CHEQUE", which is 3 (i.e: 2 for substituting Q with C, & U with K, and 1 insertion of E).

		C	H	E	Q	U	E
		1	**2**	**3**	**4**	**5**	**6**
C	**1**	0	1	2	3	4	5
H	**2**	1	0	1	2	3	4
E	**3**	2	1	0	1	2	3
C	**4**	3	2	1	1	2	3
K	**5**	4	3	2	2	2	3

Task:
Your job is to create a similarity checker software which takes user assignments as two strings, and
computes the similarity between these assignments using the LD measure expressed as a percentage. It is important to note that the software should be case sensitive where character **‘A’ ≠ ‘a’**.

The formula for determining the percentage is:

$$Similarity = 100 \cdot \frac{\left| LD - \max\left(|string1|,\ |string2|\right) \right|}{\max\left(|string1|,\ |string2|\right)}$$

Input

The first line contains T (the number of test cases). Then T test cases follow in the following format:

First line of the test case contains $|S_1|$ (size of string1).

Second line contains string1.

Third line of the test case contains $|S_2|$ (size of string2).

Fourth line contains string2.

The characters in string1 and string2 only include (A-Z, a-z, ?, ., _, !). Note that input strings will not contain spaces (or any whitespace character).

Constraints

$1 <= |S_1| <= 10000$

$1 <= |S_2| <= 10000$

Sum of $|S_1|$ and $|S_2|$ over all test cases will not exceed 20000.

Output

The output contains just one line with two numbers: one integer denoting LD value and second is a float number denoting the similarity between 2 strings. Use **ceiling function** to round your similarity value.

Sample input & output

Sample input	**Sample Output**
2 12 This_is_test 12 Test_is_This 5 Check 6 Cheque	7 42% 3 50%

PROBLEM 3: Slim Intelligent

Time limit: 2 seconds

Japanese have long appreciated the presence of life in all aspects of nature. They are very sensitive about their ecosystem and animal rights. It is common in Japan to find animal graves and memorial markers, a reflection of Japanese sensibility that does not distinguish between humans and animals but perceives both as equal elements of nature. Ms. Tian is a young researcher in the University of Tokyo, working on the ecosystem balance and animal preservation in Japan. She is using the latest drone imaging to study the diversity in wild animals.

As drone images are captured from the above, Ms. Tian has to detect and classify animals on the basis of their body shape. While observing the wild animals through drone images for a few months, she came up with an interesting observation. She observed that the slimmer an animal is, the smarter they act in the wild. She wants to share this interesting observation with her team members and supervisor.

For This purpose, she labeled the observed data by assigning some parameters to each animal. Given a list of animals, each represented by a 5-tuple containing the labeled name of animal, weight, IQ, age, and the number of tasks performed, Ms. Tian asked you to find the longest possible sequence of animals in the data with increasing weight and decreasing IQ level in descending order of smartness. The animal with the lowest weight-IQ ratio should be considered the "smartest." If there is a tie, the animal with the lowest age should be considered "smartest". If there is still a tie, the animal that has more tasks should be considered "smartest".

Input

The first line of input is an integer N (5 <= N <= 1000), representing the total number of animal entries a user wants to enter. The next N lines represent the input data by the user. The input data is a list of 5-tuples of the form (Name [string], Weight [int], IQ [int], Age [int], No. of tasks performed [int]), where each value is separated by a space. The list represents data of one animal per line. Each animal is guaranteed to be labeled with a different name.

Output

The first line should print an integer K, indicating the length of the longest found sequence. Make sure that the output sequence is in descending order of smartness and follows the following inequalities:

$$W(1) < W(2) < W(3) \ldots \ldots < W(K) \qquad \text{and} \qquad I(1) > I(2) > I(3) \ldots \ldots > I(K)$$

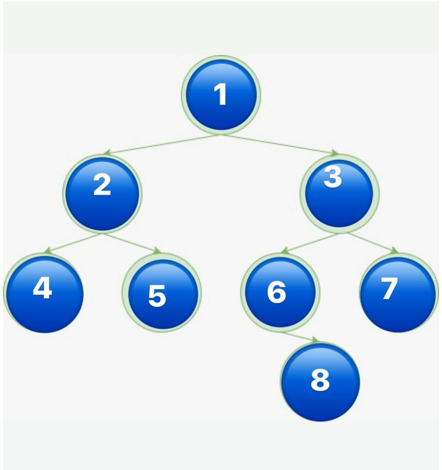
Where W is the weight and I represents the IQ of the animal. For your answer to be correct, K must be as large as possible.

Sample input	Sample Output
9 Pandy 609 130 5 3 Moky 600 210 3 2 Lino 50 200 7 5 Elo 100 400 4 7 Dony 110 300 3 6 Ballo 600 200 8 2 Gollo 800 140 6 1 Horsy 600 120 7 4 Jumpy 200 190 8 5	4
6 Alto 100 150 6 3 Caty 150 130 7 2 Jawry 110 200 9 4 Wiki 105 180 7 5 Jumpi 100 150 4 2 Flup 110 160 5 5	3

PROBLEM 4: The Arab Lineage

Time limit: 3 seconds

It was a norm in Arab history to keep record of the lineage of each tribe which still keeps up to the day. But remembering names and relations to more than two generations is a hard job. So they have created a digital family tree to store the data. So for each tribe the tree starts with the grand ancestor, as far as the tribe can remember. The grand ancestor is a father and can have two sons maximum and each of his sons can have a maximum of two sons and so on. In the first version they are just recording the most influential people of tribes, so recording just two sons at max is fine. All the people have an identity represented by numbers. Now they are interested in finding the relations between people using this tree. Given a person A and his relative B, help them to find out what is the relation of B to A.



Input

First line of the input consists of two integers, N (2 <= N <= 10^5) and C (C=N-1) where N represents the total number of people in the family and C represents the number of direct connections. The next C lines contain two integers P, Q showing that P is the direct parent of Q. The following line contains an integer M (1 <= M <= 10^5) representing the number of queries. The next M lines contain two integers consisting of Ids of person A and person B. .

Output

For each line M, print the relation of person B to person A. If both people are the same, leave the relation empty as given in the second test case.

Sample input	Sample Output
8 7 1 2 1 3 2 4 2 5 3 6 3 7 6 8 7 4 8 4 6 3 4 2 4 5 2 8 5 8 7	8 is 4's grand father's grand grand son 6 is 4's grand father's grand son 4 is 3's father's grand son 4 is 2's son 2 is 5's father 5 is 8's grand grand father's grand son 7 is 8's grand father's son
2 1 1 2 1 1 1	1 is 1's

PROBLEM 5: Real Estate Dilemma

Time limit: 3 seconds

Real estate companies are always in search of acquiring land, houses, and plazas that are adjacent to their properties. Their surveying teams move around the country to mark properties owned by them and by others. Every company is competing with each other to increase their presence by having largest areas with them which will increase their portfolio and reputation.

These companies are looking for expertise of experienced computer scientists who can develop a software for them to ease their decision process based on the information provided by the surveying team.

The surveying teams marks a property as a small case single alphabet owned by a specific owner. Each company intends to buy maximum properties and hence want to change the alphabet to theirs'. However, only a limited number of properties are available for sale. The company that can make the most consecutive properties by combining the adjacent ones will buy these available properties. At the end, the total number of maximum consecutive properties owned by a company is declared.

Example

The surveying team has given a string, "tufwfxww". Only 1 property can be purchased. Therefore, either "w" or "f" will buy the property and thus 3 will be displayed as maximum number of consecutive properties by a company.

Input

The first line of the input consists of t, ($1 \le t \le 25$) representing the total number of test cases given. Each test case is represented in two lines. The first line of each test case contains a sequence of small alphabets representing properties. The length of the sequence (n) can be $1 \le n \le 10^5$. The second line of each test case represents, maximum number of properties available for sale (k). The "k" can be a number defined as $0 \le k \le 10^5$.

Output

Output consists of t lines, each line contains exactly one number, representing the maximum number of consecutive properties after buying "n" or less number of properties.

Sample input	Sample Output
4 abab 2 kjcpckma 2 ddydyyd 1 ddydyyd 2	4 4 4 5

PROBLEM 6: Shrek's Wall

Time limit: 1.5 seconds

Shrek is an ogre that lives in swamps and hates everybody. For that reason, he has made a wall of rocks around his swamp. Every night, a fairy comes and disturbs his wall construction by shifting the stones. Shrek has to balance the wall every morning. The fairy sometimes takes away some stones, and sometimes she places extra stones on the wall, to further confuse the ogre.

Your task is to create an algorithm which can level the wall in a minimum number of stone movements. The stone configuration is given as an integer array of size N. You have to choose a value from the array as your target value and then in the minimum number of moves, make all the array elements equal to the target element. In one move, you can increment or decrement an array element by 1.

Input

First line of the input contains T, which is the number of test cases separated by lines. Each test case contains N space-separated integers denoting the values of array elements.

Output

There will be T lines in output. Each line of output will be a number indicating the minimum number of moves to make the numbers in the list equal to the target value. You can choose any number from the array as your target value, but it should ensure that you get the minimum number of moves as your answer.

Constraints

$1 <= T <= 100$

$1 <= N <= 10^6$ (it is guaranteed that sum of N over all T test cases won't exceed 10^6)

Sample input	Sample Output
2 1 2 3 1 10 2 9	2 16

PROBLEM 7: Friendly Aliens Come to GIK Institute

Time limit: 3 seconds

GIK Institute is located at the base of the scenic Tarbela dam, surrounded by picturesque natural beauty and breathtaking views. The dam itself is a stunning landmark and a source of pride for the university and the local community. The surrounding area is lush with greenery and dotted with trees, creating a peaceful and serene environment for the university campus. The body of water created by the dam adds to the overall aesthetic appeal of the university, and offers opportunities for recreational activities such as boating or fishing.

Additionally, the university has walkways, parks, or green spaces near the water where students and faculty can gather and enjoy the scenic views. The close proximity to the dam and the water makes the university a unique and attractive place to study, work, and live. The combination of the stunning natural surroundings, academic opportunities, and close-knit community makes GIKI a truly special place.

A few years ago, a friendly group of aliens from a distant planet landed on a mountain range. They have a favorite food which they bring with them called Kiti-Kata. When a group of people approach them, they choose one person to share their favorite food with. To determine the chosen person, all members of the group stand in a circle. They number the persons 1 to p. Aliens follow an elimination process to choose a person, stated below:

1. In the first step, person 1 is removed.
2. In the second step, they skip one person. Person 3 is removed.
3. In the third step, they skip two persons. Person 6 is removed.
4. And so on...
5. The last person in the circle is the chosen one.

For instance, if a group of five people visited the aliens, they would be removed in this order: 1, 3, 2, 5, and the person at position 4 would eat kiti-kata with the aliens. Similarly, if a group of four approached, they would be removed in this sequence: 1, 3, 4, and the person at position 2 would be treated to the alien's hospitality. Can you help identify the lucky person to eat kiti-kata during the next trip of aliens to the mountains?

Input

As input, you are given the number of cases t ($1 \leq t \leq 1000$) in the first line, followed by t lines. Each line contains the total number of participants p ($1 \leq p \leq 5000$).

Output

The output of the program contains n ($1 \leq n \leq 1000$) lines, where each line consists of a single number, indicating the position of the person to eat kiti-kata.

Sample input	Sample Output
5 5 7 12 15 25	4 4 8 11 13

PROBLEM 8: Predator

Time limit: 3 seconds

Human beings have always been fascinated with the existence of life on other planets. A team of ***n*** astronauts is working on a newly discovered planet in search of life. Each astronaut a has a certain power value ai What they don't know is the existence of a single predator on that planet. This predator is an evolved form of life that can optimize its function. That predator also has a certain power value in addition to **two** green power drinks and **one** blue power drink to consume. At any moment this predator can perform any one of the following actions.

1. Consume an astronaut with power strictly **less than** predator power.
2. Drink green power liquid, if there's at least one left.
3. Drink blue power liquid, if not already consumed.

When predator consumes an astronaut with power a_i, this astronaut disappears, and the power of the predator is increased by $\lfloor a_i/2 \rfloor$ (floor operation). For example, if a predator consumes an astronaut with power 7, its power increases by 3.
After consuming the green power drink, this drink disappears, and the power of the predator doubles, so it increases by 2 times.
After consuming the blue power drink, this drink disappears, and the power of the predator triples, so it increases by 3 times.
Your task is to figure out optimally how many maximum numbers of astronauts can be consumed by the predator.

Input

The first line contains an integer t — number of test cases.
The first line of each test case contains integers n ($1 \le n \le 1000$) — number of astronauts and h ($1 \le h \le 10^6$) — the initial power of the predator.
The second line of each test case contains n integers ai ($1 \le a_i \le 10^8$) — powers of astronauts.

Output

For each test case, in a separate line, print the maximum number of astronauts that a predator can consume.

Sample input	Sample Output
7 4 1 2 1 8 9 3 3 6 2 60 4 5 5 1 100 5 3 2 38 6 3 1 1 12 4 6 12 12 36 100 3 5 15 1 13	4 3 3 3 0 4 3

Explanation:
In the first case, you can proceed as follows:

1. consume *green* power drink. $h = 1 \cdot 2 = 2$
2. consume the astronaut 2. $h = 2 + \left\lfloor \frac{1}{2} \right\rfloor = 2$
3. consume *green* power drink. $h = 2 \cdot 2 = 4$
4. consume the astronaut 1. $h = 4 + \left\lfloor \frac{2}{2} \right\rfloor = 5$
5. consume *blue* power drink. $h = 5 \cdot 3 = 15$
6. consume the astronaut 3. $h = 15 + \left\lfloor \frac{8}{2} \right\rfloor = 19$
7. consume the astronaut 4. $h = 19 + \left\lfloor \frac{9}{2} \right\rfloor = 23$