



Gran Premio de México 2021 - Segunda Fecha

September 25th, 2021

Problems book

General Information

This problem set contains 11 problems; pages are numbered from 1 to 14, Without considering this page. Please, verify your book is complete.

A) Program name

- 1) Solutions written in C/C++ and Python, the filename of the source code is not significant, can be any name.
- 2) Solutions written in Java, filename should be: *problem_code.java* where *problem_code* is the uppercase letter that identifies the problem. Remember in Java the main class name and the filename must be the same.
- 3) Solutions written in Kotlin, filename should be: *problem_code.kt* where *problem_code* is the uppercase letter that identifies the problem. Remember in Kotlin the main class name and the filename must be the same.

B) Input

- 1) The input must be read from *standard input*.
- 2) The input is described using a number of lines that depends on the problem. No extra data appear in the input.
- 3) When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input.
- 4) Every line, including the last one, ends with an end-of-line mark.
- 5) The end of the input matches the end of file.

C) Output

- 1) The output must be written to *standard output*.
- 2) When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output.
- 3) Every line, including the last one, must end with an end-of-line.

Problem A

Alice Birthday

Recently Alice celebrated her birthday. Bob knows that collecting graphs is a hobby for Alice, for that reason he decided to give her an undirected graph for her birthday. The graph consisted of N nodes and M undirected edges.

Alice was playing with her new graph when she started wondering what would happen if she deleted some edges from the graph. She thinks that the number of connected components is an important property for graphs, so she would like to know how many ways of deleting some edges there are so the graph have K connected components after deleting such edges.

For all K from 1 to N , help Alice to find the number of ways of deleting edges so the graph has K connected components. Two ways of deleting edges are different if there is an edge that is deleted in a way but not in the other. It is allowed to delete all edges or not to delete any edge at all.

Input

The first line contains two integers N and M ($1 \leq N \leq 14$ and $0 \leq M \leq \frac{N(N-1)}{2}$), the number of nodes and the number of edges.

Each of the following M lines contains two integers u and v ($1 \leq u, v \leq N$ and $u \neq v$), the nodes that the edges connect. It is guaranteed that there are not multiple edges connecting a single pair of nodes.

Output

Print N lines. In the K -th line print the number of ways of deleting some edges so there are K connected components after deleting such edges. Print the answer modulo 998244353.

Input example 1 5 4 1 2 1 3 1 4 1 5	Output example 1 1 4 6 4 1
Input example 2 2 0	Output example 2 0 1
Input example 3 4 6 1 2 1 3 1 4 2 3 2 4 3 4	Output example 3 38 19 6 1

Input example 2 1 13 0	Output example 2 1 0 0 0 0 0 0 0 0 0
Input example 3 13 17 0 1 5 3 1 0 2 1 4 2 5 2 1	Output example 3 126914455 362778099 800719102 15040153 238529512 843998224 698108042 421844973 174968030

Problem C

Cut the Deck!

Alice and Bob are playing a new game with cards, the game is played with a deck which contains N cards, each card has a color blue or red, and half of the cards in the deck are blue, half are red. At the beginning of the game Alice will shuffle the deck, and will ask Bob to "cut the deck", which means to take a stack of cards off the top from the deck and place it on the bottom. For example, if a deck with 6 cards has the following configuration after shuffling: 'BRRRBB' and Bob cuts the deck at card 2, then the resulting deck will be 'RRBBBB'. Once the deck is cut by Bob, the game begins, Alice will start from the top of the deck, turning one card at a time, if in some point there are more red cards showing than blue cards, then, Alice wins the game, if Alice gets to turn all the cards from the deck, then Bob wins.

Bob insists this is unfair as he has not been able to win any of the games against Alice, he believes the secret for him to win is to properly select the card at which he has to cut the deck, so he will practice his technique shuffling a deck of cards and then selecting the card where to cut. Can you help Bob to given a deck configuration determine where he has to cut the deck in order to win the game?

Input

The first line of input contains a single integer T ($1 \leq T \leq 100$), the number of test cases. Each of the next T lines contains a string S representing the deck Bob has to cut, if the character at position i in the string is a 'B', it means the card at position i has blue color, if the character is 'R', the card has a red color. The length N ($1 \leq N \leq 10^6$) of the string will always be even, and it is guaranteed it has the same amount of 'R' and 'B'.

Output

For each test in the input print a line containing the number of cards Bob has to take from the top of the deck when cutting the deck so he can win the game, if more than one solution exists, print the one with the fewest number of cards. In case this is not possible print -1 for that case.

Input example 1	Output example 1
3	0
BRBR	1
RBRB	5
BBRRRB	

Problem D

Dislike the Raisins

There is a brand new cereal that Jaime really enjoys, but, as every good thing, this cereal has an issue, it is mixed with the only ingredient Jaime dislikes: raisins. A few days ago Jaime tried to remove all the raisins from a box of cereal and he found it was practically impossible to him. Jaime will eat the box of cereal using a spoon that can handle up to S units of cereal, he eats the box of cereal always filling the spoon until no more cereal is left in the box, in case the box is not empty and he can not fill the spoon, he will put in the spoon all the cereal left in the box. Jaime considers his spoon of cereal is an outstanding spoon if it does not contain raisins.

Jaime just noticed that the box contains information about the amount of cereal C and the amount of raisins R that are inside of the box, knowing this information, Jaime wants to find, if he was lucky enough, what is the maximum number of outstanding spoons he could have while eating the box? Similarly, if he was that unlucky, what is the minimum number of outstanding spoons he could have?

Input

The first line of input contains a single integer T ($1 \leq T \leq 1000$), the number of test cases in the input. Each of the next T lines contains three integer numbers separated by a space, C , R , and S , ($0 \leq C, R \leq 10^9$, $1 \leq S \leq 10^9$), representing, respectively, the amount of cereal, and raisins in the box, and the units the spoon can hold. You can assume the size of a cereal and a raisin is the same, so any of them takes just one unit of the spoon.

Output

For each test case in the input output a line containing two integer numbers separated by a space, the maximum and minimum number of outstanding spoons Jaime can take from the box.

Input example 1	Output example 1
3	4 0
20 10 5	4 0
20 11 5	2 1
6 2 3	

Problem E

E-13 Storage Unit

Planet E-13 orbits around a star in a faraway galaxy named UAZ. In that planet, people live for centuries and some inhabitants, such as Thanitos, like to record videos from their daily life (and oh dear, he has really made a few). Up until now, Thanitos has been using an old technology where videos are recorded in tapes that take up too much physical space, and he wants to paste some of the N videos he has made, into a new kind of storage unit that is really small but has much more capacity (a single storage unit could hold up the content of several video tapes). He has his N video tapes in chronological order, and for each of them he has calculated how much space would be required in MB to put them into a storage unit. A requirement is that the videos selected to be placed in a storage unit must be consecutive. That is, if for instance, Thanitos has the following 7 videos, and the storage unit has a 2048 MB capacity:

Video number	Size
1	512
2	17
3	802
4	10
5	1024
6	905
7	130

Then if the first video Thanitos puts in the storage unit is video number 1, he must put in the same storage unit, videos number 2, 3 and 4, since they are consecutive and still fit in the storage unit. If the first video Thanitos puts in the storage unit is Video 2, then he must also put videos 3 and 4, or if the first video Thanitos puts is video 5 then he must put video 6 (video 7 could not be put since it would exceed the capacity of the storage unit).

Thanitos wants to know 2 things, given the size of each of the N videos (in MB) and the capacity of a storage unit (possible given in MB, GB or TB):

1. What is maximum number of videos R , such that, regardless of which video is selected as the initial one to be stored (from the ones numbered 1 to $N - R + 1$), the storage unit will hold at least that number R of videos.
2. What is the least video number L , such that if it is the first video that is put in the storage unit and we try to put $R + 1$ consecutive videos in it, the storage unit capacity would be exceeded (A value of -1 indicates that no matter what video is selected as initial one, the capacity is not exceeded)

Just in case you don't remember, $1GB = 1024MB$, and $1TB = 1024GB$.

Input

The first line contains an integer number N and a string C separated by a space ($1 \leq N \leq 10^6$, $100MB \leq C \leq 10240TB$), N is the number of videos that Thanitos has made, and C is a string representing the capacity of a single storage unit. C is suffixed with an M , a G or a T indicating the capacity is in Megabytes (MB), Gigabytes (GB) or Terabytes (TB). The second line contains N integers separated by a space, where the i -th integer represents the size in Megabytes s_i the video with number i would require to be stored in the storage unit. For all i , $10 \leq s_i \leq \min(10240, C)$.

Output

Print a line containing two integers separated by a space: R and L , according to the description given above.

Input example 1	Output example 1
9 2G 512 17 802 10 1024 905 130 1838 15	2 5

Problem F

Flipped Factorization

Hugo is learning to factorize numbers into primes, and he's going to make a program to test his new algorithm. First, he will factorize every integer from 1 to N and save every factorization. Then, he will rebuild every integer in that range from its factorization, and add up all the results (the number 1 is rebuilt as 1). Of course, he will get as result $N(N+1)/2$ if his algorithm is correct.

But he made a silly mistake in the second part: suppose that he's rebuilding the integer m with prime factorization $p_1^{a_1} p_2^{a_2} \dots$, then he really will add $a_1^{p_1} a_2^{p_2} \dots$ to the answer, because he accidentally flipped the base and its exponent in every prime.

He's obtaining very strange results, and you are curious to see how far is his wrong answer from the correct one.

Input

The first line contains the positive integer N ($1 \leq N \leq 10^{14}$).

Output

Output a line containing the wrong answer obtained by flipping the bases and the exponents of every prime in every factorization. Since this number can be very large, output it modulo $10^9 + 7$.

Input example 1 5	Output example 1 8
Input example 2 10	Output example 2 28
Input example 3 20	Output example 3 66

Notes

The first 10 positive integers are rebuilt as:

- $1 \rightarrow 1$
- $2 = 2^1 \rightarrow 1^2 = 1$
- $3 = 3^1 \rightarrow 1^3 = 1$
- $4 = 2^2 \rightarrow 2^2 = 4$
- $5 = 5^1 \rightarrow 1^5 = 1$
- $6 = 2^1 \times 3^1 \rightarrow 1^2 \times 1^3 = 1$
- $7 = 7^1 \rightarrow 1^7 = 1$
- $8 = 2^3 \rightarrow 3^2 = 9$
- $9 = 3^2 \rightarrow 2^3 = 8$
- $10 = 2^1 \times 5^1 \rightarrow 1^2 \times 1^5 = 1$

Hence the answer for the second example is $1 + 1 + 1 + 4 + 1 + 1 + 1 + 9 + 8 + 1 = 28$.

Problem G

Grid of Letters

John likes to solve the word search puzzles that appear in the newspaper he reads while drinks his coffee. A word search puzzle consists of the letters of words placed in a grid, which usually has a rectangular or square shape. The objective of the game is to find and mark the words from a list in the grid, these words may be placed horizontally, vertically, or diagonally.

John has a lot of experience solving the puzzles and he finds it kind of boring now, he decided to play a different puzzle on the same grid. Instead of searching for words, he will search for the longest path of consecutive letters he can find in the grid, the path can be followed from one letter to the other if the letters are adjacent in any direction (horizontally, vertically, or diagonally). For example, if John has the following grid

A	B	H
F	F	G
B	D	E
A	F	C

The longest path of consecutive letters he can find in the grid is conformed with the letter 'CDEFGH', in the following positions:

		H
	F	G
	D	E
		C

John has told you his new puzzle idea, your response was it is even easier than word search, to demonstrate this to him, you decided to write a program that finds the length of the longest path of consecutive letters John can find in a word search grid.

Input

The first line of input contains two integer numbers separated by a space N and M ($1 \leq N, M \leq 1000$), representing the number of rows and columns in the grid. The next N lines contains a string with M characters, the j -th character of the i -th line, represents the character at the i -th line and j -th column of the grid. All characters in the grid will be uppercase letters from the english alphabet.

Output

Output a line containing a single integer, the length of the longest path of consecutive letters that can be found in the grid.

Input example 1	Output example 1
4 3 ABH FFG BDE AFC	6

Input example 2	Output example 2
3 3 AZC YBC ABC	3

Problem H

Haunted House

Haunted houses are one of the most popular Halloween attractions. Bob plans to open his own haunted house for Halloween this year. His haunted house consists of N rooms numbered from 0 to $N - 1$.

Bob has N ghosts and each of them has a unique energy value between 1 and N . A ghost with energy x can be active for x consecutive seconds, and then it needs to rest for x seconds until it can be active again. But there is one more flaw, if that ghost scares a person while it's active, then it needs to rest immediately for x seconds until it can be active again, since scaring people is tiring for ghosts.

Before deciding where to place the ghosts, Bob wants to simulate some scenarios. A scenario is defined by a permutation p_0, p_1, \dots, p_{N-1} , indicating that Bob places a ghost with energy p_i in the room i , and a sequence t_0, t_1, \dots, t_{M-1} , indicating the arriving time of people to the haunted house.

If a person arrives to the house at second t_j , then they visit the cell 0 at second t_j . When a person visits the cell i at second T , there are two cases:

1. If there is an active ghost in the room, then the person gets scared and abandons the house immediately. The ghost starts resting at second $T + 1$.
2. Otherwise, the person visits to the room $i + 1$ at second $T + 1$ or abandons the house if $i + 1 = N$.

You are given a scenario, help Bob to simulate it and find what is the room where the people get scared and the second when that happens. Note that all ghosts start being active at second 0.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 10^5$). The second line contains a permutation p_0, p_1, \dots, p_{N-1} of size N ($1 \leq p_i \leq N$), indicating that a ghost with energy p_i is placed in room i . The third line contains M integers t_0, t_1, \dots, t_{M-1} ($0 \leq t_i \leq 10^5$), the second when the people arrive to the house. All people arrive at different time.

Output

Print M lines. In the j -th line print two space-separated integers: the number of the room where the person with number $(j - 1)$ get scared and the second when they get scared. If the person abandons the house without being scared in any room, print two -1 separated by a space.

Input example 1	Output example 1
2 3	0 0
2 1	-1 -1
0 2 5	1 6

Input example 2	Output example 2
3 3	0 0
3 1 2	0 4
0 4 10	0 10

Problem I

Integer Multiplicative Persistence

In mathematics, the persistence of a number is the number of times one must apply a given operation to an integer before reaching a fixed point at which the operation no longer alters the number. The multiplicative persistence of an integer, is how often one has to replace the number by the product of its digits until one reaches a single digit. For example, the multiplicative persistence of 39 is 3, because it takes three steps to reduce 39 to a single digit, in the first step the operation is $3 * 9 = 27$, in the second step the operation is $2 * 7 = 14$, and the third and last step is $1 * 4 = 4$.

Your task for this problem is given a number N , find its multiplicative persistence.

Input

The first line of input contains a single integer T ($1 \leq T \leq 1000$), representing the number of test cases in the input. Each of the next T lines contains a single integer N ($0 \leq N \leq 10^9$), representing the value N to which the multiplicative persistence should be found.

Output

For each test case in the input output a line containing one integer number, the multiplicative persistence of N .

Input example 1	Output example 1
6	0
0	0
5	1
10	2
25	3
39	2
27	

Problem J

John in the Amusement Park

In an amusement park are A different activities where people can participate to improve their happiness. Each activity i in the park has a fixed duration d_i in minutes, and provides an amount of h_i happiness. Since the activities require some preparation they are offered to be started at certain times in the day while the park is opened. The times at which an activity can be started are given in minutes since the park opening, this is, if an activity is listed as it can be started at 10 it means the activity starts 10 minutes after the park opening.

John has been very tired from work this week and decided to go to the amusing park to get some happiness participating in certain activities, but, as next week seems to be a very complicated work week, John wants to select the activities in such way that the total happiness of participating on those activities during the day is maximized. The total happiness John can get from participating in the activities equals the sum of happiness each activity provides for participating in it. John believes that two rules should be followed to achieve maximum happiness:

- John will not start an activity until the activity he is participating finishes.
- John will not start an activity after the park closing time, but, the last activity may be finished after the park closing time.

Given the closing time of the park, the number of activities in the park, their duration and starting times, can you help John find the maximum happiness he can get from his visit to the amusement park?

Input

The first line of input contains two integer numbers separated by a space A ($1 \leq A \leq 500$) and T ($1 \leq T \leq 10^6$), representing the number of activities in the amusing park, and the minute at which the park closes. The next $2 \times A$ lines describe each of the park activities. Each activity is described in two lines of input. The first of these two lines contains three integer numbers separated by a space h_i, d_i, t_i ($1 \leq h_i \leq 1000$, $1 \leq d_i \leq T$, $1 \leq t_i \leq 10$), representing, respectively, the happiness h_i and duration d_i of the i -th activity, and the t_i number of different times when the i -th activity can be started. The second description line for an activity contains t_i integer numbers separated by a space, representing each of the times $t_{i,j}$ when the activity can be started ($0 \leq t_{i,j} < T$). It is guaranteed that the times for each activity will be given in increasing order, this is $t_{i,j} < t_{i,j+1}$ where $1 \leq j \leq t_i$.

Output

Print a single line with an integer number, the maximum number of happiness John can get in a day in the park.

Input example 1	Output example 1
3 100 40 10 3 0 40 60 100 80 2 0 20 50 15 1 1	150

Problem K

K-Binary Repetitive Numbers

A number N is called to be K -binary repetitive if it's binary representation using K bits can be represented as a shorter binary string concatenated a number of times, for example, $N = 10$ is 4-binary repetitive since it's binary representation with 4 bits (1010) can be represented as concatenating the binary string 10 to itself, but, it is not 5-binary repetitive since its binary representation using 5 bits 01010 can not be represented as a shorter binary string concatenated a number of times.

Given a number K , can you find how many different numbers N are K -binary repetitive?

Input

The first line of input contains a single integer T ($1 \leq T \leq 10^5$), the number of test cases. Each of the next T lines contains a single integer number K ($1 \leq K \leq 10^6$).

Output

For each test in the input print a line containing one integer number, the number of different N that are K -binary repetitive, since this number can be big print it modulo $10^9 + 7$.

Input example 1	Output example 1
2	0
1	2
2	