Problems Overview

Problem A: Alice in the Digital World

Problem B: Reservoir

Problem C: Terraced fields

Problem D: Message

Problem E: Directory Management

Problem F: Ultimate Pipeline Game

Problem G: Nature Reserve

Problem H: Printer Scheduling

Problem I: Divisor Game

Problem J: Swimming Balls

Problem K: ICPC Team Selection

Problem L: Olympiad Training

The input and output for all the problems are standard input and output.

Problem A: Alice in the Digital World

After returning from the Wonderland, Alice needs to improve her scientific skills in this current digital world. Alice decides to participate the ACM-ICPC Asia Nha Trang Regional Contest 2016 to evaluate her actual performance. Her most favorite problem in the contest is following.

Given an array of positive integers $A = a_1, a_2, \ldots, a_n$, a subarray A_i^j of A is a sequence of continuous elements in A, *i.e.*, $A_i^j = a_i, a_{i+1}, \ldots, a_j$ (where $1 \le i \le j \le n$). The weight of A_i^j is the sum of all its elements, *i.e.*, $\sum_{k=i}^{j} a_k$.

Given an integer m, your task is to find the maximum weight subarray of A that contains only one m as the minimum element. You can assume that A always contains at least one element with value m.

Input

The input consists of several datasets. The first line of the input contains the number of datasets, which is a positive number and is not greater than 20. The following lines describe the datasets.

Each dataset is described by the following lines:

- The first line contains 2 positive integers n and m ($n \le 10^5$; $m \le 2^6$).
- The second line contains n positive integers, each with value at most 2^6.

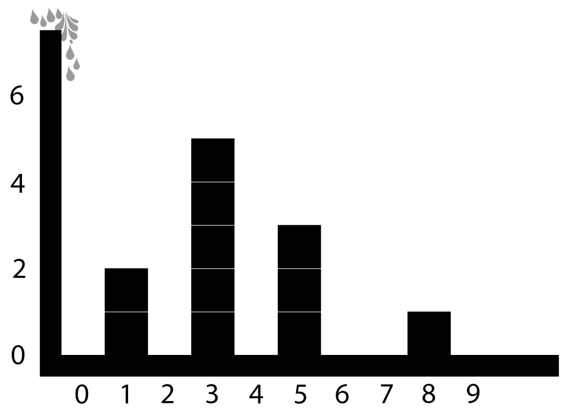
Output

For each dataset, write out on one line the found maximum weight.

Sample Input	Sample Output
1 6 2 1 3 2 6 2 4	12

Problem B: Reservoir

A big reservoir was built in Red river using a dam. Assume that the reservoir is a rectangular box with unit length width. The reservoir consists of many tanks. An example a cross section of an empty reservoir along its length and height dimensions is shown in the picture below:



Water flows in from the top left gate into the reservoir. The tanks in the resevoir are constructed using water resistant walls. Each wall is one unit length thick (along the width dimention) and has its height smaller than the height of the reservoir.

Given the location and the height of the walls and the unit volume K of water flowing in, your task is to figure out the last wall water flows over.

Input

The input consists of several datasets. The first line of the input contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset is described by the following lines:

- The first line contains one positive integer N that is the number of walls separating the tanks ($N \leq 10^5$).
- The second line contains N positive integers L_i where L_i is the horizonal location (along the length dimention of the reservoir) of the i^{th} wall ($1 \leq L_i \leq 10^9, L_i > L_{i-1} + 1$ for $i > 1$).
- The third line contains N positive integers H_i where H_i is the height in unit length of the i^{th} wall ($1 \leq H_i \leq 10^5$).
- The fourth line contains an integer Q that is a number of queries ($Q \leq 10^5$).
- In the next Q lines, each line contains a positive integer K that is the unit volume of water flowing in the reservoir ($K \leq 10^{15}$)

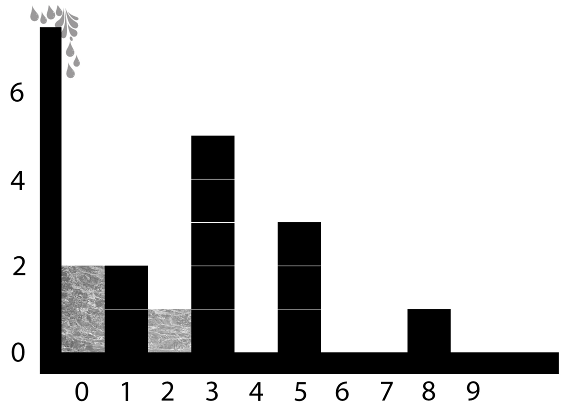
Output

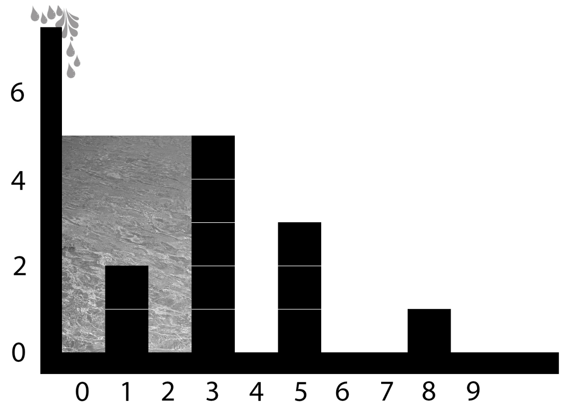
For each dataset, write out Q lines where the i^{th} line contains the index of the last wall that water flows over for the i^{th} query. If there is no wall that water flows over, output 0.

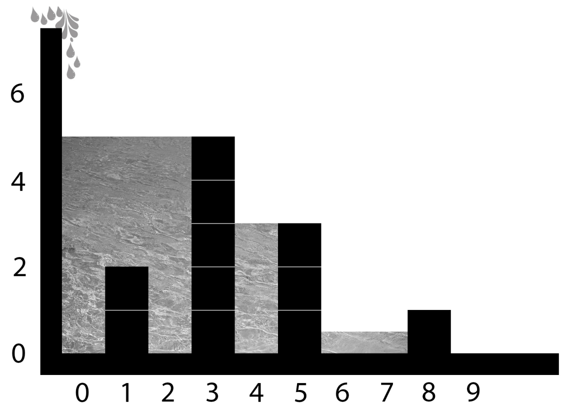
Sample Input	Sample Output

`1`
`4`
`1 3 5 8`
`2 5 3 1`
`3`
`3`
`13`
`17`
	`1`
	`1`
	`3`

Explanations:







Problem C: Terraced fields

Terraced fields with beautiful landscapes in Northwest Vietnam are popular destinations for tourists. At each terraced field selected as a tourist attraction, the local authorities build a staircase alongside the terraced field. The steps are numbered from 1 to n starting from the bottom of the hill. At steps divisible by 8 (i.e. steps numbered 8, 16, 24, etc.) and at the final step (i.e. n^{th} step), the step number is stone engraved as a height indication for tourists. It is considered that 6 and 8 are lucky digits so people used a precious stone to specifically engrave these digits.

There is a tour that goes to a terraced field having n steps. The price of the tour is the number of precious stone engraved digits on its steps.

For given n, your task is to determine the price of the tour.

Input

The input consists of several datasets. The first line of the input contains the number of datasets, which is a positive number and is not greater than 100,000. The following lines describe the datasets.

Each dataset is described by one line containing an integer n ($1 < n \leq 10^{18}$).

Output

For each dataset, write out on one line containing the price of the tour.

Sample Input	Sample Output
4	1
9	2
32	4
56	3
18	

Problem D: Message

A student wants to send to his friend a message, which is a text string p consisting of only lowercase latin alphabet letters. To encrypt his message, he creates a lowercase alphabet string h of size n that contains p as a substring. The student is curious to find out how many different ways there are to create such a string h.

Given two positive integers n, M and a string p consisting of only lowercase latin alphabet letters, let's denote K to be the total number of different ways to create a lowercase alphabet string h of size n such that p is a substring of h. Your task is to find the remainder of K divided by M.

Input

The input consists of several datasets. The first line of the input contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset is described by the following lines:

- The first line contains two positive integers n, M ($n \leq 10^{12}; M \leq 10^{12}$);
- The next line contains the text string p consisting of at most 50 lowercase latin alphabet letters.

Output

For each dataset, write in one line the remainder of K divided by M.

Sample Input	Sample Output
2 2 100 ab 3 100 ab	1 52

Problem E: Directory Management

Tired of using existing badly written operating systems, Hieu decided to write his new one. Of course, his new operating system will be awesome, bug-free, fast and easy to use. He has finished most of the work, and now he is asking you to do one last task: Implement a directory manager. Initially, Hieu's computer directory is empty. The current directory is the root directory.

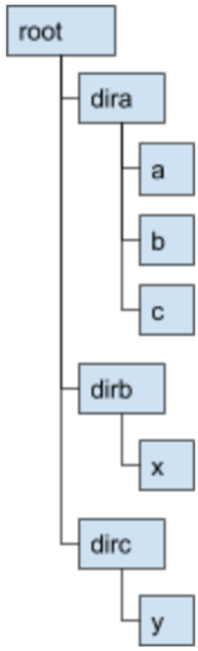


Figure 1

The directory manager keeps the directory in a rooted-tree structure. In each directory, the children are sorted in lexicographical order.

He can do one of the following actions:

- *MKDIR* ***s***: create a child directory named ***s*** inside the current directory where ***s*** is a string.
    - If the current directory already contains a child directory named ***s***, print "ERR" and do nothing.
    - Otherwise, print "OK"
- *RM* ***s***: remove a child directory named ***s*** inside the current directory where ***s*** is a string.
    - If there is no child directory named ***s***, print "ERR". Otherwise, print "OK".
- *CD* ***s***: change the current directory to a child directory named ***s*** where ***s*** is a string.
    - If ***s*** is equal to the string ".." and the current directory is the root directory, print "ERR" and do nothing.
    - If ***s*** is equal to the string ".." and the current directory is not the root directory, then you need to change the current directory to the parent directory and print "OK".
    - If there is no child directory named ***s***, print "ERR" and do nothing.
    - If there is a child directory named ***s*** then you need to change the current directory to ***s*** and print "OK".
- *SZ*: Print the total size of the current directory.
    - The size of a directory is defined as 1 + total size of its children.
- *LS*: list the child directories of the current directory in lexicographical order. For example, if you are at "root" directory of the example in Figure 1, *LS* would print like Figure 2:
    - If there is no child directory, print "EMPTY".

```
dira
dirb
dirc
```

Figure 2

    - If there are more than 10 child directories in the current directory, print the first 5 children, followed by a line containing only "...", followed

by the last 5 children.
- ○ If the number of child directories is between 1 and 10 inclusively, print all children.

- *TREE*: list all the directories inside the current directory, in the pre-order traversal where the children are visited in lexicographical order. For example, if you are at "root" directory in the above image, *TREE* would print like Figure 3:
  - ○ If there is no child directory, prints "EMPTY".
  - ○ If there are more than 10 directories (counting all directories of any level deep) inside the current directory (including the current directory), instead of printing all the lines, only print the first 5 lines, followed by a line containing "...", followed by the last 5 lines.
  - ○ If the number of child directories is between 1 and 10 inclusively, print all directories in the pre-order traversal.

```
dira
a
b
c
dirb
x
dirc
y
```

Figure 3

- *UNDO*: undo the effect of the last command that satisfy the following three conditions. If there is no command to UNDO the print "ERR". Otherwise, print "OK".
  - ○ The command has to be one of the following commands: *MKDIR* or *RM* or *CD*.
  - ○ The command did not result in printing "ERR".
  - ○ The command has not yet been undone by any UNDO command.

Given a list of commands, your task is to execute those commands and find out the printed output.

Input

The input consists of several datasets. The first line of the input contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset is described by the following lines:

- The first line contains only one integer Q – the number of commands.
- The next Q line contains one of the commands described above.
  - ○ For commands *MKDIR*, *RM*, *CD*: ***s*** can contains only lowercase characters (except for the case "CD ..") and the length of ***s*** does not exceed 3.

Each dataset has the following constraints:

- The total number of commands does not exceed 100,000.
- The total number of *MKDIR* and *RM* commands does not exceed 5000.

Output

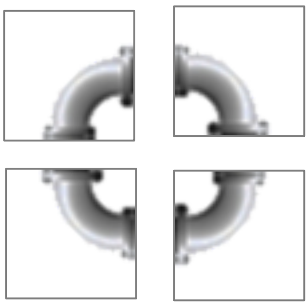
The output for each dataset should be separated by an empty line. For each command, you need to print out exactly like the above explanations.

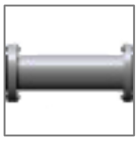
Sample Input	Sample Output
1 22 MKDIR dira CD dirb CD dira MKDIR a MKDIR b MKDIR c CD .. MKDIR dirb CD dirb MKDIR x CD .. MKDIR dirc CD dirc MKDIR y CD .. SZ LS TREE RM dira TREE UNDO TREE	OK ERR OK OK OK OK OK OK OK OK OK OK OK OK OK 9 dira dirb dirc root dira a b c dirb x dirc y OK root dirb x dirc y OK root dira a b c dirb x dirc y

Problem F: Ultimate Pipe Game

Pipe games are interesting and hard puzzle games. In these games, our mission is to connect the pipes to make the water flow in the pipeline without leaking out from a source to a destination. Today we are playing a new generation of pipe game, the Ultimate Pipe Game.

In this game, we have a grid of m rows and n columns. Cells on the grid are either empty or blocked. The cell at row i and column j is denoted as cell (i, j). We only put pipes on empty cells and each cell can contain only one pipe. There are 3 kinds of pipes: curved pipes, horizontal pipes and vertical pipes. Note that a curved pipe can be rotated as shown in the picture below, but we cannot rotate a horizontal pipe to make a vertical pipe or vice versa.





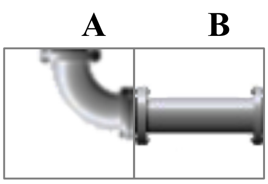
Horizontal pipe **Vertical pipe**

Curved pipe which can be rotated

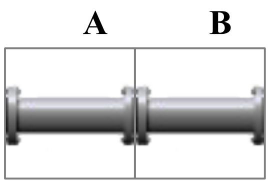
By putting pipes on empty cells, water from a cell can travel to its adjacent cell if and only if the pipes in two cells are connected by their heads.

Our mission in this Ultimate Pipe Game is to place pipes on the empty cells with following requirements:

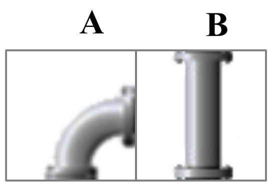
- We need to put pipes on **every** empty cell and each empty cell must contain exactly one pipe.
- We have to make sure water in the pipeline will not leak out. Starting from any empty cell with a pipe, assuming that we have water flows out from that pipe in one of the two directions, water can travel through the pipeline to other cells without leaking out and return to the starting cell. In this case, we call it a **cycle pipeline.**
- We can have multiple disjoint **cycle pipelines**.



A and B are connected

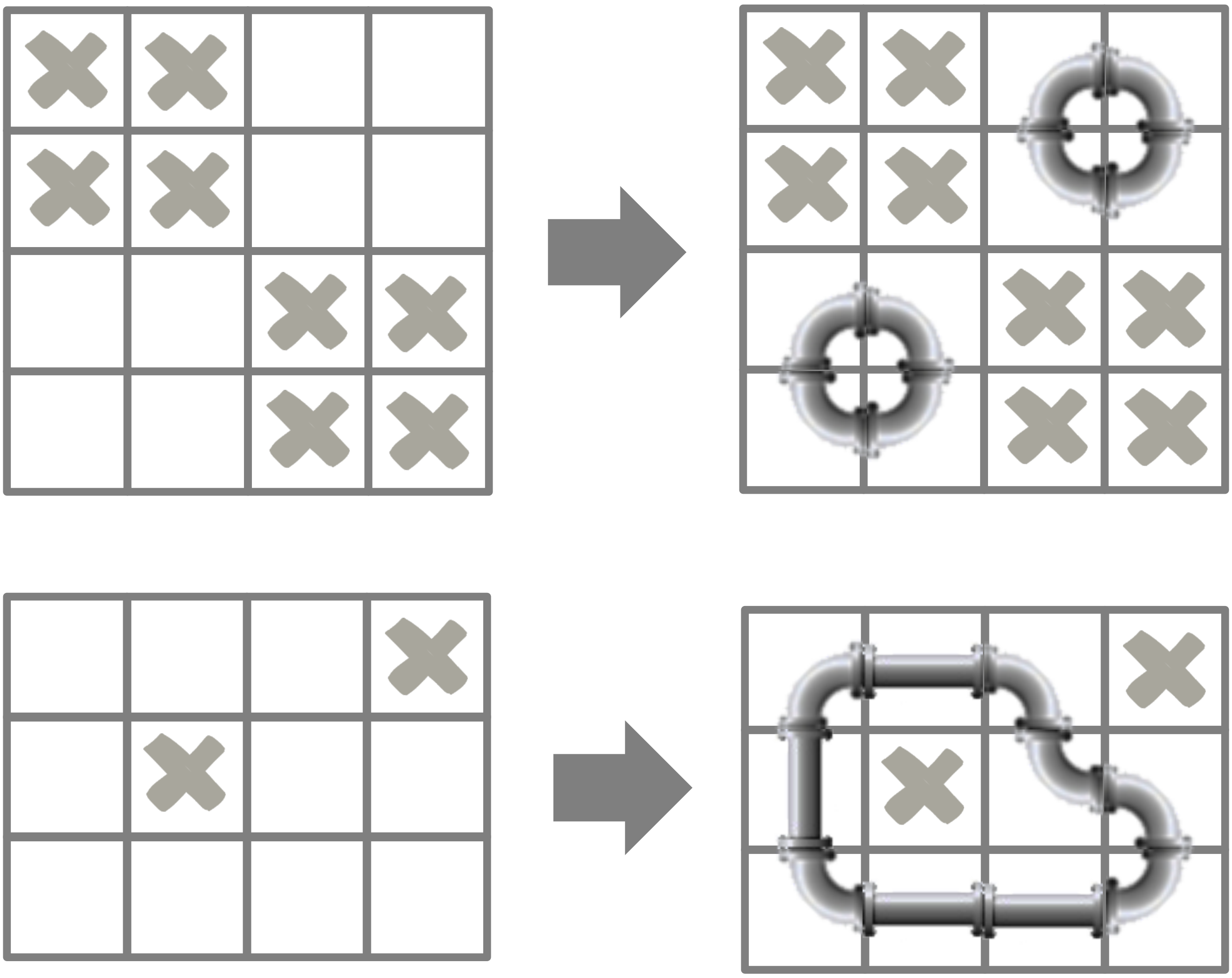


A and B are connected



A and B are not connected

- Putting a horizontal pipe and a vertical pipe at cell (i, j) costs $h_{i,j}$ and $v_{i,j}$ coins, respectively. Putting in a curved pipe is free.



Your task is to find a way, if exists, to put pipes on empty cells to minimize the total number of coins. You can assume that there are unlimited number of pipes.

Input

The input consists of several datasets. The first line of the input contains the number of datasets, which is a positive number and is not greater than 100. The following lines describe the datasets.

Each dataset is described by the following lines:

- First line contains two integers m and n ($2 \le m, n \le 20$).
- The next m lines describe the grid where each line contains n characters. The j^{th} character at i^{th} line denotes the state of cell (i, j): '.' for an empty cell or '#' for a blocked cell.
- In the next m lines, each line contains n integers denoting the cost of putting a horizontal pipe on a cell. The j^{th} number on the i^{th} line is an integer $h_{i,j}$ ($0 \le h_{i,j} \le 100$) where $h_{i,j}$ is 0 for a blocked cell.
- In the last m lines, each line contains n integers denoting the cost of putting a vertical pipe on a cell. The j^{th} number on the i^{th} line is an integer $v_{i,j}$ ($0 \le v_{i,j} \le 100$) where $v_{i,j}$ is 0 for a blocked cell.

Output

For each dataset, write out the result in the following format.

- If you can find a way to put pipes on empty cells to fulfill the requirements, write out on one line the string "YES y" where y is the minimum total number of coins we need to pay.
- Otherwise, write out on one line the string "NO".

Sample Input	Sample Output
3	YES 0
4 4	YES 10
##..	NO
##..	
..##	
..##	
0 0 1 2	
0 0 3 0	
1 2 0 0	
2 3 0 0	
0 0 1 2	
0 0 3 0	
1 2 0 0	
2 3 0 0	
3 4	
...#	
.#..	
....	
1 2 3 0	
4 0 1 2	
3 1 2 3	
3 2 1 0	
5 0 2 2	
3 1 2 3	
3 3	
...	
...	
...	
0 0 0	
0 0 0	
0 0 0	
1 1 1	
1 1 1	
1 1 1	

Problem G: Nature Reserve

In a Nature Reserve and Wildlife Park, there are N environmental monitoring stations to monitor temperature, atmospheric pressure, humidity, fire, water quality, etc. Each station, labeled from 1 to N, uses solar panels to self-supply energy for its operations. There is a communication network consisting of several 2-way communication channels between pairs of stations. All stations are connected via this communication network.

To process data at each station, the Nature Reserve and Wildlife Park needs to install a Smart Data Analysis program (with the size of L bytes) to all environmental monitoring stations. The program is initially installed directly to S stations, then broadcasted to and installed in all other stations via the communication network.

To save energy, all communication channels are initially in an idle state and its needs to be activated to send information. It takes E_{ij} energy units to activate the communication channel between station i and station j. Once a channel is activated, it takes one energy unit to transmit one byte via this channel.

Your task is to determine the minimum energy units required to send the Smart Data Analysis program to all stations from the initial S stations.

Input

The input consists of several data sets. The first line of the input contains the number of data sets, which is a positive number and is not bigger than 20. The following lines describe the data sets.

Each data set is described by the following lines:

- The first line contains four positive integers: the number of environmental monitoring stations N, the number of 2-way communication channels M, the size of the program L (in bytes), and the number of initial stations S $\left(1 \le S \le N \le 10^4, 1 \le M \le 10^6, M \le \frac{N(N-1)}{2}, 1 \le L \le 10^6\right)$.
- The second lines contain S positive integer representing the initial S stations.
- Each of the following M lines contains three positive integers i, j and E_{ij} to denote that there is a 2-way communication channel between station i and station j, and it takes E_{ij} energy units to activate this channel ($E_{ij} \le 10^6$).

Output

For each data set, write out on one line the minimum energy units required to send the Smart Data Analysis program to all stations from the initial S stations.

Sample Input	Sample Output
1 4 6 10 1 3 1 2 4 1 3 8 1 4 1 2 3 2 2 4 5 3 4 20	37

Problem H: Printer Scheduling

There are n files to be printed using m identical printers. The files are numbered from 1 to n. The printers are numbered from 1 to m. Assuming each page takes one unit of time to print, for each file i, we have the following information:

- The number of pages it contains, p_i (i.e. the time it takes to print file i);
- Ready time r_i (the printing of file i cannot be started before time r_i);
- Finish time d_i (the printing for file i has to be completed no later than time d_i).

We can assume that $d_i - r_i \geq p_i$, $i = 1, 2, ..., n$. The printing process of a file can be interrupted between pages. In other words, while printing file f, the printer can interrupt this job and move to print a different file. The printing process of file f can be resumed on any available printer afterwards. We can assume that:

- The time it takes to move the printing of a file from one printer to another printer is negligible.
- The starting time for printing the files is 0.

A schedule of printing n files using m printers has to satisfy the following requirements:

- The printing of each file j cannot be started before the ready time r_j;
- The printing of each file j has to be completed no later than the finish time d_j;
- At any one time, the printing of file j can be processed by at most one printer and the total amount of printing time of file j, i.e. its number of pages, is p_j;
- At any one time, each printer can only process at most one page of one file.

Your task is to find if there exists a schedule to print n files using m printers satisfying the requirements.

Input

The input consists of several datasets. The first line of the input contains the number of datasets, which is a positive number and is not greater than 100. The following lines describe the datasets.

Each dataset is described by the following lines:

- The first line contains 2 integers n, m ($1 \leq n, m \leq 200$);
- The i^{th} line in the following n line contains 3 positive integers p_i, r_i, d_i ($p_i, r_i, d_i \leq 30{,}000$ for $i = 1, 2, ..., n$).

Output

For each dataset, write out on one line the string "YES" if there exists a schedule and "NO" otherwise.

Sample Input	Sample Output
1 4 2 4 2 7 3 3 8 3 4 7 5 1 10	YES

Problem I: Divisor Game

RR and Flash are playing a game with a list of numbers that are distinct initially. In this game, two players will take alternative turns. In each turn, a player can select a number X in the list and replace it with number D if D is a divisor of X and D is smaller than X.

For example, with the list (1, 3, 12), the valid moves are:

- replace 12 with one of the following numbers: 1, 2, 3, 4 or 6;
- replace 3 with 1.

The player who takes the last move will lose the game. At the beginning, Flash will select a non-empty initial list of distinct numbers between A and B inclusively and RR is the one who makes the first move.

Your task is to calculate the number of possible ways for Flash to select the initial list where he can be sure that he would win the game assuming both players play optimally.

Input

The input consists of several datasets. The first line of the input contains the number of datasets, which is a positive number and is not greater than 100. The following lines describe the datasets.

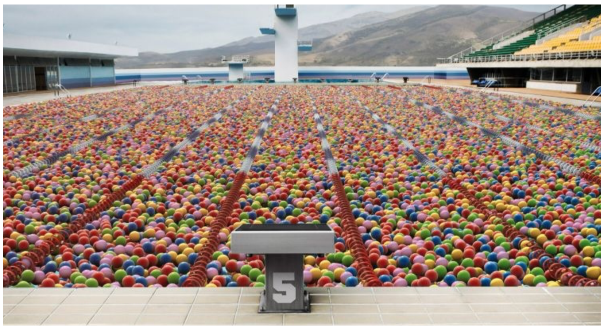
Each dataset is described by a single line that contains 2 integers A and B ($1 < A \leq B \leq 10^{12}$, $B\text{-}A \leq 10^5$).

Output

For each dataset, write out on one line the result modulo 10^9+7.

Sample Input	Sample Output
4 2 4 2 5 2 6 2 7	2 4 8 16

Problem J: Swimming Balls



To attract more people coming to the pool, the manager puts a number of swimming balls into the pool. He wants to calculate the level of water in the pool after putting *V* m^3 of water and *n* given balls into the pool. The i^{th} ball is a homogeneous sphere (mass is distributed evenly) having a specific weight (or unit weight) w_i and a radius r_i. The pool is a rectangle with dimensions *W* (width) x *L* (length) x *D* (depth) in meters. Your task is to calculate the level of water (measured in meters to the bottom of the pool) after putting *V* m^3 of water and *n* given balls into the pool. This level of water can be a most *D*.

We assume that the specific weight of water is 1.0, there is no interaction among the *n* balls, and the Archimedes' principle is perfectly guaranteed.

Hints: the volume of a homogeneous sphere having a radius r could be computed using the following formula: $V = \frac{4}{3}\pi r^3, \pi = 3.141592653589793$.

Input

The input consists of several datasets. The first line of the input contains the number of datasets, which is a positive number and is not greater than 50. The following lines describe the datasets.

Each dataset contains the following information:

- The first line contains five integers *n, W, L, D, V* $(1 \leq n \leq 10^3, 2 \leq W, L \leq 10^3, D \leq 5, V \leq W.L.D)$
- The i^{th} line of the *n* following lines contains two positive floats r_i and w_i indicating the specific weight and radius of the i^{th} ball $(0 < w_i, r_i \leq 2)$.

Output

For each dataset, write in one line the number indicating the water level with the precision to 3 decimal places, which should be at most *D*.

Sample Input	Sample Output
2 1 2 2 2 5 1.0 1.0 1 2 2 2 5 0.8 1.0	2.000 1.786

Problem K: ICPC Team Selection

The coach of Nha Trang University – Mr Van has just organized a contest to form its ICPC teams. There was $3N$ students attending the contest. The i^{th} student scored P_i in the contest.

The coach wants to form N different teams, each team has 3 students, to take part in the regional contest based on this result. In his experience, the performance of a team usually equals to the median of team members' individual result (i.e. the result of the second-best student).

The coach wants to maximize S - the sum of his N teams' performance. Your task is to calculate S.

Input

The input consists of several datasets. The first line of the input contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset is described by the following lines:

- The first line contains a positive integer N ($N \leq 100$).
- The second line contains $3N$ positive integers $P_1, P_2, \ldots, P_{3N}$ ($P_i \leq 100$).

Output

For each dataset, write out on one line the value S.

Sample Input	Sample Output
1 2 8 8 6 9 10 9	17

Explanation: one way to form a team is

- team 1: student 1, student 2, student 3
- team 2: student 4, student 5, student 6

Problem L: Olympiad Training

Beside the pretigious ACM-ICPC contest, Nha Trang University also hosted the Vietnamese Collegiate Olympiads in Informatics 2016. This Olympiad has several events where students compete individually to show their skills and knowledge.

After a successful contest this year, N junior students in Nha Trang University expressed their interests to join the team next year to represent their university. But in order for them to be at a medal contender level, the coach, Mr. Van, has to teach them M topics. The i-th student will require a_{ij} minutes to understand the j-th topic. Teaching K students $X_1, X_2, \dots, X_K$ the j-th topic will require $max\ (a_{X_1,j}, a_{X_2,j}, \dots, a_{X_k,j})$ minutes and teaching them all M topics will require $\sum_{j=1}^{M} max\ (a_{X_1,j}, a_{X_2,j}, \dots, a_{X_k,j})$ minutes.

Given K - the number of students in a group to train, your task is to help Mr. Van decide who he should pick in order to minimize the total time he needs to teach them all M topics.

Input

The input consists of several datasets. The first line of the input contains the number of datasets which is a positive integer and is not greater than 100. The following lines describe the datasets.

Each dataset is described by the following lines:

- The first line contains 3 integers N, M, Q $(1 \le Q \le N \le 20, M \le 10000)$.
- The i-th line of the next N lines contains M integers a_{ij} $(0 \le a_{ij} \le 10^9)$.
- The u-th line of next Q lines contains a query which is an integer K $(1 \le K \le N)$ representing the number of students in the group that Mr. Van needs to train.

Output

For each dataset, write out Q lines where the i^{th} line contains the minimal time required for the i^{th} query.

Sample Input	Sample Output
2 2 2 1 1 3 3 2 1 3 3 3 1 4 9 2 6 3 3 5 5 1 2 3	4 11 14 18