Algorithms for Programming Contests
SS20 - Week 9

Chair for Foundations of Software Reliability and Theoretical Computer Science,
TU München

Mikhail Raskin, Tobias Meggendorfer, Stefan Jaax, A.R.Balasubramanian,
Christoph Welzel

Welcome to our practical course! This problem set is due by

Wednesday, 07.07.2020, 6:00 a.m.

Try to solve all the problems and submit them at

`https://judge.in.tum.de/conpra/`

This week's problems are:

The following amount of points will be awarded for solving the problems.

Problem	A	B	C	D	E	F*
Difficulty	easy	easy	medium	medium	hard	medium
Points	4	4	6	6	8	6*

* catchup problem / bonus points, do not count towards total

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

Problem A
Contact List

A few days ago, Lea experienced one of the horrors of modern life: She dropped her smartphone. Now, her screen is cracked and sometimes random locations on the screen act as if they had just been pressed. When sending a message to one of her contacts, she enters the name of the contact into a searchbox. If the name matches exactly, she can send the message with just another click. However, now that her screen is cracked, this means that sometimes her phone already sends the message to “Bob”, while Lea meant for it to be sent to “Bobby”, which are totally different people. This has embarassed Lea quite a few times now, so she wants to rename some of her contacts such that no contact is a prefix of another one. Can you tell her how many contacts she has to rename?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case consists of an integer n, the amount of contacts Lea has in her phone. n lines follow, each line containing the name of a contact (where the first letter is in “A” to “Z” and the rest is in “a” to “z”).

Output

For each test case, output one line containing “Case #i: x” where i is its number, starting at 1, and x is the minimal amount of contacts Lea has to rename. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 10000$
- Contact names are unique.
- Contact names are not longer than 500 characters.

Sample Input 1

```
1
7
Bob
Bobby
Boba
Charles
Charly
Julia
Julian
```

Sample Output 1

```
Case #1: 2
```

Sample Input 2

```
7
4
Bfugw
Ksdb
Ctg
Bfug

3
Pgqh
Mlvo
Pgqhzot

7
Opmp
Faokkia
Fao
Opmpn
Qkqv
Qewyu
Faos

3
Ct
Qxhu
Qxhuzr

8
Olp
Wafgmp
Olpt
Wafgm
Olpv
Wbgl
Wbglhlq
Waf

4
Alna
Al
Nl
Mmybw

8
Wlyppv
Etdtfz
Wl
Wly
Etdtf
Etdtfzu
Spwaw
Aogja
```

Sample Output 2

```
Case #1: 1
Case #2: 1
Case #3: 2
Case #4: 1
Case #5: 4
Case #6: 1
Case #7: 4
```

Problem B
J&J's

Lea and Bea have arranged for some friends to come over for a big LAN party lasting the whole weekend. Since healthy food is not appropriate for LAN parties, they bought tons of pizza, lasagna, and of course even more snacks like gummybears, chocolate cookies, and J&J's.

Unfortunately, the moment they wanted to start their gaming session, the switch decided to retire, leaving them with no working local network to play on. After some hours of trying to get the switch to work again, they gave up and decided to play the infamous no-LAN-tonight-game called "Toss dem J&J's". The game works as follows: One person takes a few J&J's and tosses them towards a bunch of glasses that are standing in a line. The J&J's have to be tossed in such a way that every glass that gets some new J&J's gets the same number of new J&J's. Also if two non-neighbored glasses get new J&J's, all glasses in between also need to get the same number of new J&J's. If Lea messes up and tosses two J&J's into one glass and only one J&J into another one, she is forced to remove the J&J's she tossed into both glasses. Obviously, she loses the game instantly.

If Lea did not violate any of the previous rules, she scores points equal to the total number of J&J's in all glasses she hit. Therefore, it is crucial to know how many J&J's are in a given interval of glasses. Can you help Lea keep track of the number of J&J's in the glasses during the game?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with two integers n k, where n is the number of glasses available and k is the number of queries. k more lines follow. Each has one of the following forms:

- q a: Return the number of J&J's in glass a.
- i ℓ r v: Add v J&J's to each glass in the interval ℓ to r (including ℓ and r).

Output

For each test case output one line containing "Case #i: x" where i is its number, starting at 1, $x = (r_1 + \cdots + r_j) \bmod 1000000007$ and $r_1 \ldots r_j$ are the answers to the glass queries.

Constraints

- $1 \leq t \leq 5$
- $1 \leq k \leq 10^5$
- $1 \leq n \leq 10^6$
- $1 \leq v \leq 1000$
- $1 \leq a \leq n$
- $1 \leq l \leq r \leq n$
- There is always at least one sum-query.

Sample Input 1

```
3
4 3
i 1 3 1
q 3
q 1

6 5
i 2 5 2
i 1 6 3
q 3
i 1 3 1
q 1

5 2
i 1 5 4
q 2
```

Sample Output 1

```
Case #1: 2
Case #2: 9
Case #3: 4
```

Sample Input 2

```
5
4 7
i 1 2 645
i 2 4 945
q 3
q 2
q 1
i 1 3 606
q 1

5 6
i 2 5 695
i 3 5 863
q 3
q 2
q 3
q 4

8 7
i 3 6 488
i 4 6 234
q 1
i 5 8 179
q 3
i 4 8 930
q 8

4 3
i 2 3 519
q 2
q 2

4 8
q 2
i 1 3 664
i 2 3 545
i 2 4 724
q 4
i 1 3 305
q 2
q 2
```

Sample Output 2

```
Case #1: 4431
Case #2: 5369
Case #3: 1597
Case #4: 1038
Case #5: 5200
```

This page is intentionally left blank.

Problem C
Broken Tetris

As a kid, Lea loved playing Tetris on her Gameboy. Since then, her favorite console had to endure a lot, from being thrown ragefully into various corners to tears of joy from beating the highscore. Unfortunately, this left a lot of marks such that some features of the game are no longer available:

- Pieces can no longer be rotated.
- Pieces can no longer be moved.
- Pieces spawn at random horizontal positions.
- Only rectangular pieces can spawn.
- The size of the playing field varies from game to game.
- Full rows are no longer erased.

Since Tetris was her favorite game growing up, she cannot help but turn the game on every once in a while and just watch the pieces fall down. As the playing field can get so huge, that it is no longer clear which point of the field is the highest, Lea wants you to write a program that shows the height of the highest position on the playing field at any point during the game. Can you help her out?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case begins with two integers n and k. n is the width of the playing field, k is the total number of blocks to spawn. k lines follow describing the blocks. Each line contains three integers w, h and p. w and h give the width and height of the block, p the offset from the left side of the playing field, i.e. $p = 0$ means that the block spawns at the leftmost position of the playing field.

Output

For each test case output one line containing "Case #i: $r_1 \dots r_k$" where i is its number, starting at 1, and r_j is the maximum height of the stacked blocks on the playing field after the j-th block has landed.

Constraints

- $1 \leq t \leq 5$
- $2 \leq n \leq 10^6$
- $1 \leq k \leq 10^5$
- $1 \leq w, h \leq 100$
- $0 \leq p < p + w \leq n$

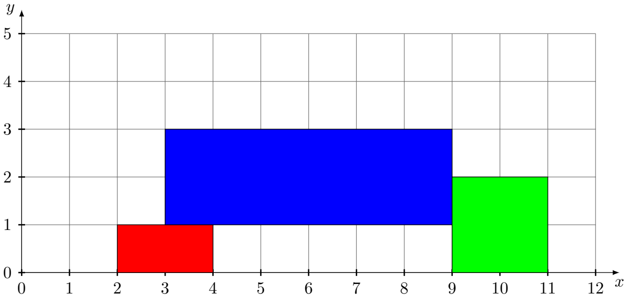

Figure C.1: Visualization of first sample input

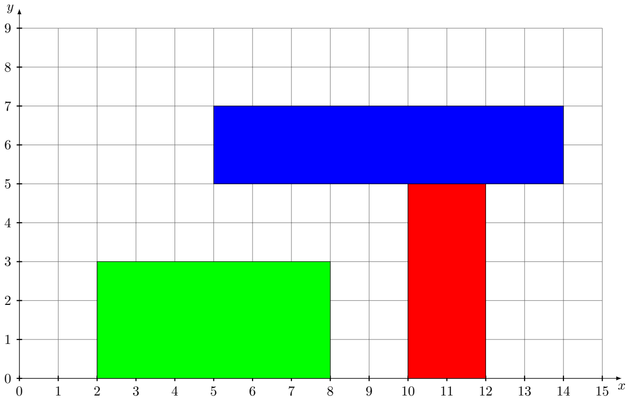

Figure C.2: Visualization of second sample input

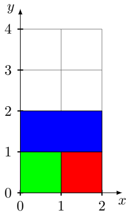

Figure C.3: Visualization of third sample input

Sample Input 1

```
3
12 3
2 1 2
6 2 3
2 2 9

15 4
2 5 10
6 3 2
9 2 5
1 1 1

2 3
1 1 1
1 1 0
2 1 0
```

Sample Output 1

```
Case #1: 1 3 3
Case #2: 5 5 7 7
Case #3: 1 1 2
```

Sample Input 2

```
4
7 9
3 4 2
4 4 3
4 2 1
1 2 4
4 4 2
3 3 2
5 2 1
1 3 2
3 1 1

13 10
3 2 3
3 2 4
4 5 4
1 5 6
4 2 0
2 5 4
2 2 8
5 2 7
4 3 5
5 5 3

12 3
4 3 8
3 4 6
2 3 9

13 9
3 4 4
1 5 3
4 2 2
2 1 7
3 4 4
5 4 4
4 3 5
1 3 11
1 2 0
```

Sample Output 2

```
Case #1: 4 8 10 12 16 19 21 24 25
Case #2: 2 4 9 14 14 14 14 14 17 22
Case #3: 3 7 7
Case #4: 4 5 7 7 11 15 18 18 18
```

This page is intentionally left blank.

Problem D
Treehouse

During summer break, Lea and her friends love to climb trees. The bigger the tree, the more fun it is!

This summer, they explored the nearest forest and climbed the biggest tree they could find. Then, each of them built a treehouse on the spot they liked most and settled in to relax. After a while, Lea wanted to visit all of her friends' treehouses to see how they built them and have a look at all the glorious treehouses.

She already planned a scenic route through the whole tree to have a look at all the treehouses from different angles so she can decide who has the coolest treehouse. Can you tell her how far she has to climb to complete the entire route?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case starts with single line containing n, the number of branching points of the tree, indexed from 1 to n. Branching point 1 is the root of the tree, down on the ground. Lea starts in branching point 1 and ends in the last node of the tour. If she has not come back to branching point 1, Lea will spend the night in one of the treehouses she visits.

n lines follow, with the i-th line containing an integer c_i followed by c_i integers $b_{i,j}$ indicating that there are tree branches connecting branching point i to branching point $b_{i,j}$.

A single line follows containing an integer v followed by v integers v_i indicating which branching points Lea wants to visit.

Output

For each test case, print a line containing "Case #i: x" where i is its number, starting at 1 and x is the minimum number of branches Lea needs to climb along to visit all v branching points in the order specified. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 5000$
- $0 \leq c_i \leq (n-1)$
- $1 \leq b_{i,j} \leq n$
- The tree is connected and has $n-1$ branches.
- $1 \leq v \leq 100000$
- $1 \leq v_i \leq n$

Sample explanation

In the first sample, case 1 there are 5 branching points in the tree. Lea starts on the ground, so she has to climb 3 branches to get to branching point 5. Then, she climbs back down to 1, again along 3 branches. Now, she climbs up to branching point 4 (along 2 branches) and then to branching point 3 (along another 2 branches). Thus the result is $3+3+2+2=10$.

Sample Input 1

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

6
2 3 5
0
1 6
0
2 2 4
0
6 3 6 3 5 2 4
```

Sample Output 1

```
Case #1: 10
Case #2: 8
```

Problem E
Ghost

One lazy afternoon, Lea and her friend Bea sit at home. They are both bored out of their minds, so they decide to play a game to pass the time. Since both are too lazy to get up from the sofa and get a boardgame from the cupboard, they settle on the game “Ghost”. “Ghost” is a simple word game, where Lea and Bea take turns announcing a single letter to build a word fragment. The rules are as follows: The announced letters have to form the beginning of a valid word. However, the player who completes a word loses.

So for example if Lea starts with “P” and Bea replies with “i”, then Lea would lose if she said “g” or “e”. She could, however, say “z” (starting to spell “Pizza”, for example) and thus continue the game. However, if Lea (or Bea, respectively) did not know any word starting with “Piy”, then announcing “y” would be against the rules.

Knowing this day would come, Lea has already prepared a list of all the words she knows. Smugly, she assumes that Bea does not know any words she does not know herself. Now, Lea and Bea play a few rounds culminating in a “final showdown” round. (All of those rounds are independent of each other, so two rounds could spell out the exact same word.) The loser of the last round has to get up and get a boardgame from the cupboard. Bea is also quite skilled at the game, so Lea wants to know if she can win against Bea even if Bea plays perfectly.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case consists of two integers n and w, with n being the amount of rounds played and w being the amount of words Lea knows. w lines follow, each containing a word s, consisting of lowercase letters from “a” to “z”.

Output

For each test case, output one line containing “Case #i:” where i is its number, starting at 1. Then output four lines, corresponding to four different scenarios. In the i-th line, print “victory”, if Lea can win the last round n according to the i-th scenario, and “defeat” otherwise.

- Scenario 1: Lea begins in round 1. The winner of round k begins in the next round $k + 1$.
- Scenario 2: Lea begins in round 1. The loser of round k begins in the next round $k + 1$.
- Scenario 3: Bea begins in round 1. The winner of round k begins in the next round $k + 1$.
- Scenario 4: Bea begins in round 1. The loser of round k begins in the next round $k + 1$.

Constraints

- $1 \leq t \leq 50$
- $1 \leq n \leq 1000$
- $1 \leq w \leq 10000$
- $1 \leq |s| \leq 300$

Sample Input 1

```
2
2 5
chilling
wicked
haunting
spooky
fortress

4 2
piece
pies
```

Sample Output 1

```
Case #1:
victory
defeat
defeat
victory
Case #2:
defeat
defeat
victory
victory
```

Sample Input 2

```
5
4 12
yyhib
yyiel
yywfo
yypgj
yyirkx
yylixp
yzgjux
yyrlr
yyird
yylca
yyrdug
yyiihb

1 3
zgo
zon
zeg

3 7
zhoph
zfigualf
zfiwfq
zfigul
zwbjg
zfiwr
zfiguaw

4 6
zvmnj
zvmnwox
zvsae
zvcttb
zvsacfk
zgujn

4 3
ztswwxr
ztubkb
ztswb
```

Sample Output 2

```
Case #1:
defeat
defeat
victory
victory
Case #2:
defeat
defeat
victory
victory
Case #3:
defeat
defeat
victory
victory
Case #4:
victory
defeat
defeat
victory
Case #5:
victory
victory
defeat
defeat
```

Problem F
Catmon Go

Lea is playing a new game on her smartphone called **Catmon Go** (**Cr**e**a**tures **t**hat **m**ake **o**dd **n**oises). In this game, creatures called Catmon appear, or spawn, randomly at given locations. These locations are arranged in a line from leftmost to rightmost.

Lea can go to a certain location and catch all the Catmon currently in her view, which extends a certain amount to the left and to the right of her current position. Weirdly, this amount changes randomly as the game progresses. Catching the Catmon in her current view adds them to Lea's in-game inventory and makes them disappear from the view.

In addition to these two events, the Catmon can also disappear on their own, or despawn, after a certain time. If the Catmon has been caught in between spawning and despawning, nothing happens and it still considered caught, however otherwise the Catmon is gone and cannot be caught any more.

Lea played this game the whole day. At the end of the day, she wanted to see how many Catmon she had caught today. Unfortunately, the server connection was broken at that point, so she was unable to see her inventory and that number. However, she managed to extract the event logs for that day from the game. This gives her all the spawn events, despawn events and catch events that happened, including their locations and views.

Can you help Lea replay the events of the day and tell her how many Catmon she has caught?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing two integers n k, where n is the number of locations and k is the number of events to follow. The locations are numbered from 1 through n. k more lines follow. Each is of one of the following forms:

- `s` a: A Catmon spawns at location a, increasing the number of Catmon at location a by one.
- `d` a: A Catmon despawns at location a, reducing the number of Catmon at location a by one if there currently is at least one Catmon at that location.
- `c` l r: Lea catches all the Catmon in the interval l to r (including l and r). The caught Catmon are added to Lea's inventory and the number of Catmon in that interval is reduced to zero.

A despawn event at location a happens only after a spawn event at location a, however there may be a catch event including that location in between.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is number of Catmon Lea has caught after all k events have occurred in the given order. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 5$
- $1 \leq n \leq 10^6$
- $1 \leq k \leq 10^5$
- $1 \leq a \leq n$
- $1 \leq l \leq r \leq n$

Sample Input 1

```
3
3 5
s 1
s 3
c 1 2
d 3
c 2 3

4 10
s 1
s 2
s 3
s 3
c 1 2
d 3
d 2
d 1
c 3 4
d 3

10 9
s 1
s 5
s 8
c 2 7
d 8
c 5 8
d 1
c 1 5
d 5
```

Sample Output 1

```
Case #1: 1
Case #2: 3
Case #3: 1
```

Sample Input 2

```
3
6 9
s 3
s 1
c 1 3
c 5 6
s 2
s 3
d 1
c 1 2
c 2 4

10 10
s 7
c 3 7
d 7
s 1
c 2 6
d 1
s 7
c 3 7
d 7
s 5

8 10
s 1
s 3
s 4
s 1
d 1
d 3
s 6
s 4
d 4
c 1 5
```

Sample Output 2

```
Case #1: 4
Case #2: 2
Case #3: 2
```