2019 SBU ACM Programming Selection Contest (Sponsors: Google & GSO)

(Stony Brook University, Sep 19, 2019)

Welcome to the contest! We would like to remind you of several things:

- This contest is a local ACM-style contest. Each participant must compete alone (this is not a team contest).
- The contest lasts for 3 hours (7:30PM - 10:30PM).
- C / C++ / Java / Python would be supported. GCC 7.3.0, G++ 7.3.0, Java 8, PyPy 2.7 and PyPy 3.5 are used.
- All submissions read from stdin and output to stdout.
- You can write and test your code directly on the HackerRank platform. If you have any question, don?t hesitate to ask. Again, enjoy the contest!

Fridge Magnets

You live in an apartment with a group of people that were formerly your friends. All such friendships have long since turned to animosity due to petty roommate disputes stemming from inadequate communication. While rekindling any friendly feelings is a lost cause, the situation is so tense that you can't shake the feeling that it's tethering on the breakout of violence; talking face to face is out of the question.

Fridge magnets to the rescue! The only way for you to leave cleaning reminders that actually get read is to plop them with large letters in a place where one can't make excuses about having missed them!

Your nearby corner-store sells sets of Alphabet Magnets. **Each set consists of one copy of each of the 26 lowercase letters in the English alphabet.**

If running short, you realise that **the letters 'u' and 'n' may be used interchangeably**

You would like to know how many sets of Alphabet Magnets you ought to buy. (It's essential to not have to get more later because having each roommate pitch in their due of the price is already hard enough when confined to one transaction!)

In order to determine this, you've compiled a list of sample expressions, and would like to know how many sets are required to encode each expression individually.

Input Format

The first line consists of a single integer N, the number of expressions on your list. The following N lines each consist of a expression string S. Each character of S will be a lowercase letter or space.

Constraints

$1 \leq N \leq 10^5$

$1 \leq |S| \leq 100$

Output Format

The output should consist of N lines, each containing a single integer, representing the number of sets required to required to represent the corresponding expression.

Note that spaces can be represented without magnets.

Sample Input 0

```
2
no man knew you
unusual
```

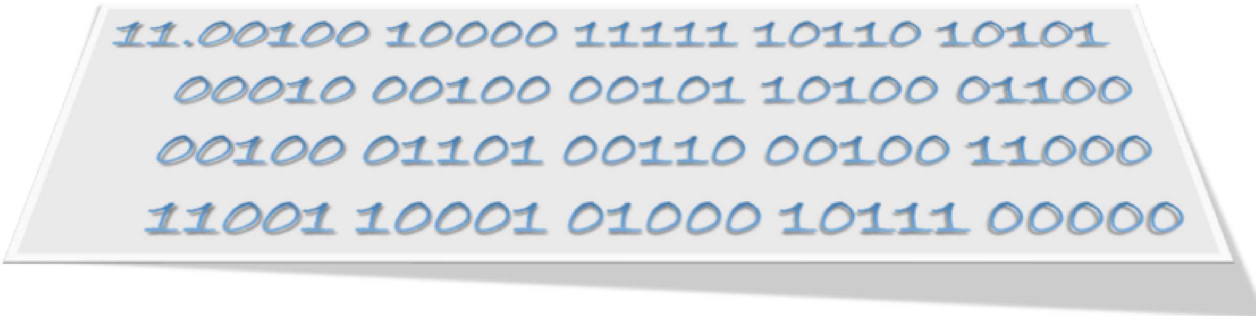
Sample Output 0

```
2
2
```

Explanation 0

In the first case, there are 3 n's and 1 u. This can be done with two sets. In the second case, there are 3 u's and 1 n. This can still be done with only two sets.

Xordan and Πper



The first 102 bits of π

Many people are aware that there is a record for most digits of π memorized, but few are aware of the Stony Brook binary π memorization contest, in which people must memorize digits of π in its binary expansion. Πper is attempting to break the world record, and keeps a notepad containing a binary substring of π every day to memorize. At the start of each day, he prints out the substring and carries it with him throughout the day to memorize.

One day, before Πper printed out the substring for the day, he needed to use the bathroom. During that time, Baby Xordan got on Πper's computer and began to play with the keyboard. Unfortunately, Πper had the file with a binary substring of length N of π open. Baby Xordan loves patterns and ended up tapping away at the keyboard in a predictable way. He first pressed the 1 key, N times, and pressed X, which XORed the original string with a string consisting of N ones. He then pressed 0 and 1 alternatively in a total of N times, and again pressed X, which XORed the modified string with a string of alternating 0s and 1s. He then pressed 001 (i.e., two 0s followed by a 1) repeatedly, and again pressed X, which XORed the further modified string with a zero string with every 3rd bit set to 1. In general, in the i^{th} iteration, Baby Xordan XORed the current state of the original string with a string, in which every bit was a zero except for every i^{th} bit, which was set to 1. He repeated this process N times in total. At the end, there was still a binary string of length N, but was now altered after a series of XORs. At this point, Baby Xordan went off the computer.

Πper came back and quickly printed out the binary substring without any suspicion of changes. When he came back later that day, he was shocked to see Baby Xordan, tampering with his computer. He realized that Baby Xordan might have tampered with the substring he printed out in the morning. He quickly checked with an official source for π, and was dismayed that the string he had printed out was not correct. Πper wants to find out how many bits were altered so that he can correct his printout.

Input Format

The first line contains N, 1 < N < 2000 , the size of the string and number of XORs. The second line contains Πper's original bit string consisting of 0's and 1's.

Constraints

$1 < N < 2000$

Output Format

A single integer representing the number of bits altered.

Sample Input 0

```
3
100
```

Sample Output 0

```
1
```

Explanation 0

Let $N = 3$, and let 100 be the original bit string. The 3 bit strings Baby Xordan will XOR with this string are 111, 010, and 001. We get the following steps:

100 ^ 111 = 011

011 ^ 010 = 001

001 ^ 001 = 000

In the end, the string is now 000, which is different from 100 in the first bit position. Thus, 1 bit was altered.

Jericho Turnpike

HackerRank

[*Copyright of the above image is held by OpenStreetMaps contributors and Wikimedia Maps.*]

New York State Route 25 (also called Jericho Turnpike along most of its breadth), is a highway originating in Manhattan and running though the entirety of Long Island, terminating at Orient Point.

You might not have been aware of this road if you don't have a car, and thus drive along it regularly. You however, are aware of it, as you have a car, and socf at people who don't! (Note: The problem setter does not have a car.)

Being the central automotive nexus of Long Island, all of your favourite stores, such as Stop & Shop, Best Buy and Home Depot, have several locations along the route. Your current lease is about to expire and, tired of your unscrupulous landlord, intend to find a new place. You would like to move somewhere along the route such that the maximum distance that you need to drive to get to any of your favourite stores is minimal.

You would like to find twice the value of this minimal distance - this represents the shortest breadth of road that contains at least one branch of all of your favorite stores.

Input Format

The first line consists of two space-separated integers K and N, where K is the number of brands to which you must have access, and N is the total number of locations of such stores along the road. You've encoded your data so that the stores are denoted by integers in the range $[0, K-1]$. The following N lines each consist of two space-separated integers, X and S, where X represents a position along the route and S denotes a kind of store.

Constraints

$0 < N < 10^8$

$0 < K \leq 100$

$-10^9 < X < 10^9$

$0 \leq S < K$

Output Format

The output should consist of a single integer, representing the shortest breadth of road that contains at least one branch of all of your favourite stores. This is to be followed by a new line character.

Sample Input 0

```
2 4
-500 0
-400 1
20 1
900 0
```

Sample Output 0

```
100
```

Dish Jenga

Jenn Gaddis is a Jenga master. For those unaware, Jenga is a game of physical skill in which there is an initial stable tower of blocks that players slowly remove blocks from until it becomes too unstable to maintain itself. In the game, a player chooses a block from within the structure, removes it, and places it on top in such a way that the tower should not crumble. As a master, Jenn Gaddis has never caused the tower to fall and wants to practice her skills with her own more challenging variant called Dish Jenga.

In Dish Jenga, Jenn Gaddis first takes N dishes with various sizes and shuffles them into a single tower. She wishes to sort the dishes in the tower by the size of each dish, with the largest dish at the bottom and smallest dish on top. To sort, the only operation he allows herself to do is to pull a dish from anywhere within the stack (without the stack falling) and to place it on top of the stack. Jenn Gaddis wishes to move as few dishes as possible. What is the minimum number of dishes she needs to move to sort the dishes in the tower?

Input Format

The first line will contain an integer N, the number of dishes in the tower. The following line will contain N integers R_i $(1 \leq i \leq N)$ representing the radii of the dishes. Every two consecutive radius values in that line will be seperated by a single space, and the same radius value may appear multiple times.

Constraints

$1 \leq N \leq 1000$

$1 \leq R_i \leq 10^9$

Output Format

A single integer representing the smallest number of moves needed to sort the tower.

Sample Input 0

```
4
2 4 7 5
```

Sample Output 0

```
2
```

Explanation 0

If Jen Gaddis had 4 dishes with the following radius in order from bottom to top, 2, 4, 7, 5, Jen Gaddis would only need to move 2 dishes. She would do this by first moving the dish with weight 4 to the top, and then moving the dish with radius 2 to the top, resulting in a stack of 7, 5, 4, 2.

Not all who wander, wander away from ice-cream

I'm in a bit of a pickle - one of potentially inadequate sweetness!

You surely know the experience of wandering about a city with friends - the buildings being a canvas for the conversation, and the aimless physical movement being necessary to keep minds from aimlessly wandering off of the wandering path of conversation.

For I have known them all already, known them all:
Have known the evenings, mornings, afternoons,
I have measured out my life with coffee spoons.
[That's T.S. Elliot for your cultural edification.]

I've thus wandered with my friends, on innumerable occasions, and come to the remarkable discovery that their 'aimless wandering' is actually a remarkably consistent fallback to operating on groupthink autopilot! When we wander the rectangular streets of Manhattan, our movement is described by a choice, at each intersection we reach, to travel North, East, South, or West, and I've discovered that we make the exact same sequence of choices on every single one of our outings!

Occasionally, with gentle nudging, I am able to send my group off in a different direction than they would typically take. Doing so replaces that choice in our route, but has no effect on any subsequent choices made.

Now, I'm especially fond of a particular ice-cream truck near Penn Station, located at the intersection 33rd and 8th. I want, nay need, to eat ice cream from there on as many occasions as our wandering will permit. (i.e. I would like to return to this intersection as many times as possible.)

Today we'll convene wandering at the intersection containing the food truck.

To avoid being excessively pushy, I would like to redirect our group on up to a certain number of occasions.

Would you please help me? Given the number of modifications that I'll be allowed, and list of our default choices, would you please tell me the maximum number of times that I'll be able to visit my favourite ice-cream truck?

You can assume the streets of Manhattan as a infinite grid of (x, y) plane and the coordinate of the ice-cream truck is at $(0, 0)$. The sequence of movements are described by the letters N, E, S, or W.

Input Format

Each input will consist of single word M specifying the sequence of movement and a non-negative integer K, the maximum number of modification that is allowed.

Constraints

Each character of M will be N, E, S, or W.

$2 \leq |M| \leq 300$

$0 \leq K \leq |M|$

Output Format

Print a single integer, the maximum number I can return to the coordinate (0, 0) [i.e: to the ice-cream truck].

Sample Input 0

```
NNWEEWWW 1
```

Sample Output 0

```
3
```

Explanation 0

By changing the the first N to a S, you can return to the ice-cream truck 3 times. (Changing the second N to a S also allows to return 3 times to the ice-cream truck.)

Sample Input 1

```
NWSE 2
```

Sample Output 1

```
2
```

Taggart Transcontinental

HackerRank

You are Dagny Taggart, one of the last remaining champions of capitalism in a world overrun by looters and moochers.

You, in fact, operate the high-speed railroad Taggart Transcontinental, while your Commie brother, the CEO, ineptly blocks many decisions that better the company on the basis of a misguided moral convictions.

Unfortunately, your brother requires you to build lines connecting every city in your country, entirely ignoring whether such lines are profitable - as though you owe rail access to those leeches!

Following the tradition laid out by your late father, the trains in your service have never once run slow. It's imperative that all your trains travel at high speeds! If but a single branch of your network can't support fast traveling trains, then inevitably your company will fester and rot! (You tacitly ignore any matters of distance; only the maximum speed, that your network can guarantee on any line, matters.)

Your subordinates give you a description of various railroad segments, each connecting two different cities. Unfortunately, some of these segments were built using the steel of communal state-owned manufacturing plants, as opposed to the scientifically and entrepreneurially brilliant product of Hank Rearden. Thus, each rail segment has an associated maximum speed with which trains may travel. Similarly, these rail segments have varying maintenance costs.

Being a business, you must operate within a budget.

Your task is to select which rail segments will form your transcontinental rail network. Such a network must service every city, and have a maintenance cost that stays within the budgetary constraint. You would like to know the maximal speed with which trains would be able to travel on any segment of your railroad.

Input Format

The first line consists of three space-separated integers, N, E, and M, where N is the number of cities, E is the number of railroad segments, and M is the budget. Your subordinates prepared the data such that the cities are denoted by integers in the range [0, N-1]. The following E lines each consist of four space-separated integers, A, B, S, and C, where A and B represent cities, S is the maximum speed that a train may travel on the railroad segment between these cities, and C is the maintenance cost for this rail segment.

Constraints

$0 < N < 10^5$

$0 \leq E \leq min(N(N-1)/2, 2 \times 10^5)$

$0 \leq M < 10^9$

$0 \leq A < B < N$

$0 < S \leq 10^9$

$0 \leq C < 10^6$

Further, no pair (A, B) will occur more than once.

Output Format

The output should consist of a single integer representing the maximum speed which can be attained on all segments of a railroad that connects every city and has a total maintenance cost less than or equal to M, provided that this is possible. If it is not possible to form a network connecting all cities while remaining within the budget, output "IMPOSSIBLE" (without quotation marks). In either case, the output should end with a new line.

Sample Input 0

```
7 12 6
1 7 5 1
1 6 7 2
1 2 4 4
1 4 6 1
2 6 5 1
2 4 6 5
2 5 8 3
3 4 5 2
3 5 5 1
3 7 7 1
4 6 6 1
5 6 4 2
```

Sample Output 0

```
5
```

Convergence

Suppose $w[1\ldots n, 1\ldots n]$ is an $n \times n$ matrix with exactly m finite nonzero entries. Every other entry $w[i,j]$ is either 0 (if $i = j$) or ∞ (if $i \neq j$). Assume that $w[i,j] = w[j,i]$ holds for $\forall i, j \in [1, n]$.

The following pseudocode shows a mysterious algorithm for updating w.

```
for k from 1 to ∞ do
    for i from 1 to n do
        for j from 1 to n do
            q = ((k − 1) mod n) + 1
            w[i,j] = min(w[i,j], w[i,q] + w[q,j])
```

It is known that as k approaches ∞, each entry $w[i,j]$ of w converges to some value $\omega_{i,j}$. Let $p[i,j]$ be the smallest non-negative integer such that when $k = p[i,j]$, $w[i,j] = \omega_{i,j}$. Your task is to compute the sum $\sum_{i=1}^{n}\sum_{j=1}^{n} p[i,j]$. Since the answer may be large, simply output the sum modulo $998,244,353$.

Input Format

The first line is an integer T $(1 \leq T \leq 30)$ indicating the number of test cases to follow.

For each test case, the first line includes two integers $n, m(1 \leq n \leq 1000, 0 \leq m \leq 2100)$. Each of the next m lines includes three integers $i, j, w[i,j]$, where $1 \leq i, j \leq n$ and $1 \leq w[i,j] \leq 10^9$.

Constraints

$1 \leq T \leq 30$

$1 \leq n \leq 1000$

$0 \leq m \leq 2100$

$1 \leq i, j \leq n$

$1 \leq w[i,j] \leq 10^9$

Output Format

Output T lines such that the answer to the i^{th} test case appears on the i^{th} line.

Sample Input 0

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

Sample Output 0

```
4
```

Explanation 0

The initial state of matrix w:

$0\,1\,\infty$
$1\,0\,1$
$\infty\,1\,0$

State of w after iteration $k = 1$:

$0\,1\,\infty$
$1\,0\,1$
$\infty\,1\,0$

State of w after iteration $k = 2$:

$0\,1\,2$
$1\,0\,1$
$2\,1\,0$

p matrix:

$0\,0\,2$
$0\,0\,0$
$2\,0\,0$

Sample Input 1

```
1
4 4
1 2 1
2 3 1
3 4 1
4 1 1
```

Sample Output 1

```
6
```

Explanation 1

The initial state of matrix w:

$0\,1\,\infty\,1$
$1\,0\,1\,\infty$
$\infty\,1\,0\,1$
$1\,\infty\,1\,0$

State of w after iteration $k = 1$:

$0\,1\,\infty\,1$

$1\ 0\ 1\ 2$
$\infty\ 1\ 0\ 1$
$1\ 2\ 1\ 0$

State of w after iteration $k = 2$:
$0\ 1\ 2\ 1$
$1\ 0\ 1\ 2$
$2\ 1\ 0\ 1$
$1\ 2\ 1\ 0$

p matrix:
$0\ 0\ 2\ 0$
$0\ 0\ 0\ 1$
$2\ 0\ 0\ 0$
$0\ 1\ 0\ 0$