Frozen 2017

Advanced Table of Contents/Verification Sheet

Team Number:_____

School:_____

#	Name	Short Name	Times Wrong (-5)	Solved (Y/N) (+60)
1	Library of Babel	babel		
2	Chess	chess		
3	Highly Composite Numbers	composite		
4	Fun with Polygons	polygons		
5	E	e		
6	Communication	communication		
7	Spawn of Enarc	enarc		
8	Pumba	pumba		
9	Eight Detection	eight		
10	Puya's Mom	mom		
11	Hashing Bashing	hashing		
12	Palindromes	palindromes		
			Total Score:	

Frozen Eagle 2017

Problem 1 **The Library of Babel** **File: babel.in**

Problem Description

The Library of Babel is a library that contains every possible book of 410 pages, 40 lines per page, and 80 characters per page. Of course, such a library couldn't exist in the real world. It would be way too big to fit in the observable universe.

However, it could exist in our computers. The trick is to not make the Library of Babel, but to make an algorithm that generates certain sections of the Library of Babel. For this problem we will be using a simple example. The library will be one very long string, and the character set will be lowercase letters and underscores. The library is generated like this: first, set the library equal to an empty string. Now, generate every string of length one and order it lexicographically, concatenate all the words and concatenate the result to the end of the library. Do the same for strings of length two, three, and so on (until 1312000). The beginning of the library looks like this:

```
_abcdefghijklmnopqrstuvwxyz___a_b_c_d_e_f_g_h_i_j_k_l_m_n_o_p_q_r_s_t
```

A library with only the character set `abc` would start like this (spaces are put between words):

```
a b c aa ab ac ba bb bc ca cb cc aaa aab aac aba abb abc aca acb acc
```

Your challenge is this: given a string (less than 100 characters) that is in the library, find the position of the the first character in the string. If the string appears multiple times, print the earliest occurrence.

Input: The first line begins with a single integer T ($1 \le T \le 50$), denoting the number of test cases. T lines follow, each with one string that is in the library.

Output: For each test case, print a line containing the position of the first character of the string within the library.

Sample Input:	5 a z efg babel kcals_is_the_best_messaging_system
Sample Output:	1 26 5 7607148 707915092789082624605212734265121524780325363810 63

Frozen Eagle 2017

Problem 2 **Chess** **File: chess.in**

Problem Description

Siaam is once again trying to beat someone in chess instead of doing his labs in CS 3. Responsible students, diligently doing their labs, are sometimes curious and look over to see how Siaam's games are going. Of course, if the game has progressed far enough, he is losing, but everyone wants to know by how much.

A quick way of figuring out how much better a player is doing than an opponent in a game is to count material. Each chess piece has a number of material points associated with it. Summing the material value of all a player's pieces gives a material score. The advantage of a player over an opponent is considered to be the difference of the material scores. For example, if Siaam has 20 points of material and his opponent has 25, the opponent has a material advantage of 5 points.

The board is described by a 8 by 8 matrix of characters, where each character describes a piece on the board. All uppercase characters belong to white. All lowercase characters belong to black. The piece descriptions are below.

Char	`.`	`P/p`	`N/n`	`B/b`	`R/r`	`Q/q`	`K/k`
Piece	empty	pawn	knight	bishop	rook	queen	king
Value	0	1	3	3	5	9	0

Given a board set-up, determine the material advantage Siaam's opponent has.

Input: The first line begins with a single integer *T*, denoting the number of data sets. Each data set contains 8 lines of 8 characters each, denoting the chessboard, as described above, followed by a line containing only "`#`".

Output: For each test case, print a line containing "`Siaam is down` ***x*** `point(s) in material.`", where *x* is the opponent's material advantage, if the material score is not tied, or "`We are still in the opening!`" if the score is tied. Remember that Siaam can play as either black or white, but he is never up material.

The sample data is on the next page.

Sample Input:	2 ......k. .... K....... NPP..... B....... .....rqq # r.bqkb.r pppp.ppp ..n..n.. .B..p... ....P... .....N.. PPPP.PPP RNBQ.RK. #
Sample Output:	Siaam is down 15 point(s) in material. We are still in the opening!

Frozen Eagle 2017

Problem 3 **Highly Composite Numbers** **File: composite.in**

Problem Description

A highly composite number is an integer with more divisors than any integer less than it. For example, 6 is highly composite because it has 4 divisors (1, 2, 3, and 6), which is more divisors than 5, 4, 3, 2, or 1. By convention, 1 is the first highly composite number. The first few highly composite numbers are 1, 2, 4, 6, 12, 24, 36, 48, 60…

Write a program that prints the number of divisors of the *n*th highly composite number mod 10^9+7.

Input: The first line begins with a single integer T ($3 \leq T \leq 3$), denoting the number of test cases. T lines follow, each with a single integer denoting n. Notice how the judge's input is given. Hard-coding the program is recommended.

Output: For each test case, print a line containing the number of divisors of the *n*th highly composite number mod 10^9+7.

Sample Input:	3 1 4 10
Sample Output:	1 4 16

Judge's Input:	3 48 120 360

Yes, we did give you the judge's input. We intentionally put it here. We *want* you to have the judge's input. Good luck!

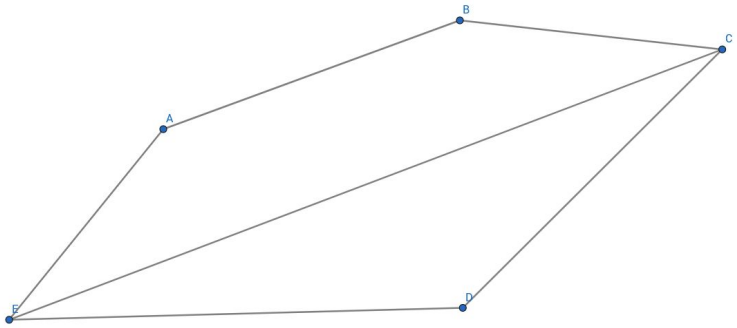
Frozen Eagle 2017

Problem 4 **Fun with Polygons** **File: polygons.in**

Problem Description

One of the golden rules for a competitive programming packet is that no team should sweep the packet by solving every single problem. Sometimes it's difficult to achieve that goal when really strong teams show up to your competition. Problems that are supposed to prevent a team from sweeping are often referred to as anti-sweepers. Also pretty often, anti-sweepers are geometry problems. So here you go, have fun!

You will be given a not necessarily convex polygon with vertices in counterclockwise order. From this you need to calculate the longest line segment that can fit entirely within the polygon. Not too hard right?



Input: The first line begins with a single integer T ($1 \le T \le 50$), indicating the number of datasets. Each dataset contains an integer V ($1 \le V \le 250$), indicating the number of vertices on the polygon and the next V lines each contain two integers X and Y which are the coordinates of a single vertex. ($1 \le X, Y \le 10000$).

Output: For each test case, print a line containing the length of the longest line segment within the polygon rounded to 4 decimal places.

Sample Input:	**Sample Output:**
1 6 0 0 50 0 100 10 100 70 80 110 40 70	136.0147

Frozen Eagle 2017

Problem 5 **E** **File: e.in**

Problem Description

The mathematical constant *e* is one that comes up very often in many fields pertaining to math or science. For example, if you put 1 dollar in a bank that offered 100% annual interest compounded continuously, at the end of one year, you would have exactly *e* dollars. *e* is an irrational number, so no one will ever know all of its digits (although you may try), but there is a rather neat formula that you can use to approximate *e*:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + \frac{1}{1} + \frac{1}{1 \times 2} + \frac{1}{1 \times 2 \times 3} + \frac{1}{1 \times 2 \times 3 \times 4} + \ldots$$

Using this formula, we can find that $e \approx 2.71828182846$. Now, write a program that, given *n*, will find the *n*th digit of *e*.

Input: The first line begins with a single integer T ($1 \leq T \leq 50$), denoting the number of testcases. T lines follow, each containing an integer N. Assume that ($1 \leq N \leq 10000$).

Output: For each test case, print a line containing the Nth digit of *e*.

Sample Input:	5 1 2 3 4 1000
Sample Output:	2 7 1 8 5

Frozen Eagle 2017

Problem 6 **Communication** **File: communication.in**

Problem Description

There is an online team communication software called Kcals. In Kcals, there are multiple channels (each with a case-sensitive alphanumeric name) where one can post messages of text. Kcals allows one to post messages which include @ links to other users (an @ sign followed by a username), and every link sends a notification to the user being mentioned. All usernames are alphanumeric and case-sensitive. Kcals keeps track of how many unread mentions a user has. Every time a user is mentioned, this number increases. Every time a user reads a channel, all the mentions previously made in that channel are no longer included in the unread mentions quote. Emmanuel (a.k.a. @latterization) wants to track his unread mentions over a period of time. Note a valid mention is the string @*latterization* ended by the end of the string or a non alphanumeric character (letters of either case, digits, and underscores are all alphanumeric).

He will give you a text file with a number of lines. Each line is either a statement or a query, and this is determined by the first character in the line. The following table outlines each command Emmanuel will give you. Italicized text in "Line format" indicates variable text.

Line format	Instruction
`#`*`channel`* `@`*`user`*`:` *`message`*	Indicates *`user`* said *`message`* in *`channel`*. Every @latterization in *`message`* increases the unread mention count of *`channel`*.
`!`*`channel`*	Indicates that Emmanuel has marked every message in *`channel`* as read, meaning that every mention made in *`channel`* prior to this command no longer contributes to the unread mention count.
`?`	Print the number of unread mentions Emmanuel has
`-`	Signifies the end of the input.

Input: The text file described above. The final line will always be "`-`".

Output: For each '`?`', print a line containing the number of unread mentions Emmanuel has at that point.

The sample data is on the next page.

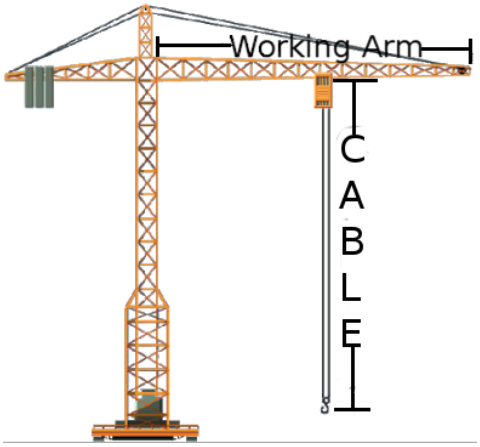
Sample Input:	? #help @JamsRamen: Just some sample data. @latterization ? #help @JamsRamen: Good luck on the judge data! #random @LamoreauxAJ: @latterization @latterization #random @JamsRamen: @latterization @latterization ? !random ? #random @latterization: Yes? #random @LamoreauxAJ: It worked! We summoned @latterization! #problemwriting @gammison: @channel I'll be disappointed if there's no donger theorem ***LINE WRAPS*** ? #general @puyag: @latterization how is college going? !general #general @latterization: good. #general @vo: @latterization please do not crowd #general. #general @JamsRamen: @latterization go to #random. #general @darkness6969: Yes, please move this to #random. #general @literally_everyone: Don't crowd #general! !help ? -
Sample Output:	0 1 5 1 2 3

Frozen Eagle 2017

Problem 7 **The Spawn of Enarc** **File: enarc.in**

Problem Description

Enarc is the legendary crane of construction. After decades of being in the trade, Enarc has learned quite a bit about concepts of trigonometry. Due to his eyes being so precise, he is always able to keep track of the location of the hook at the bottom of the cable relative to his eyes. His eyes are at the start of the working arm, so he has a very good vantage point. However, this is not the case for his offspring. The Spawn of Enarc must be fed the exact angle of depreciation from his eyes by an outside source in order to keep track of the hook properly. Enarc doesn't particularly care for his mentally challenged offspring, so he will leave that responsibility to you.



Input: The first line begins with a single integer T ($1 \le T \le 50$), denoting the number of test cases in the input. The next T lines contain three integers A, B, and C ($1 \le A, B, C \le 1000$), ($B < A$). The integer A is the length of the working arm, B is the distance from the end of the working arm to the cable trolley, and C is the length of cable hanging from the trolley.

Output: For each test case, output the angle of depreciation θ rounded to the second decimal place, in degrees, from the start of the working arm to the bottom of the cable.

Sample Input:	3 361 110 115 832 60 350 493 198 603
Sample Output:	24.62 24.39 63.93

Frozen Eagle 2017

Problem 8 **Pumba** **File: pumba.in**

Problem Description

Pumba is always late to first period calculus class because he spends too long brushing his hair in the morning. Help him arrive on time during the passing period so Mr. G doesn't give him a late grade on his test (yes it did happen). There is one issue however, because students like to walk annoyingly slow, he has to decide whether it is better to wait one second, or walk through an empty spot in the hallway which takes zero seconds. Can you help him find the shortest path assuming he can only walk horizontally or vertically?

Input: The first line begins with a single integer T ($1 \le T \le 50$), denoting the number of test cases. Each test case has two integers denoting R rows *and* C columns ($1 \le R, C \le 1000$) in the matrix, followed by R rows and C columns filled only with the characters '`S`', '`#`', '`.`', and '`E`'. '`S`' represents the position of Pumba. '`E`' represents the position of Mr. Gilliland's Calculus class, which is Pumba's goal. '`#`' represents a square with a student, while '`.`' represents an empty walkable tile. There will always be a single '`S`' and '`E`'.

Output: For each test case, print a line containing the time it would take Pumba to get to calculus if he takes the optimal path.

Sample Input:	3 7 5 S.... ####. ...#. .###. .#... .#### ....E 1 3 S.E 4 9 ....#######. .#S###E#. .#######.
Sample Output:	1 0 2

Frozen Eagle 2017

Problem 9 **Eight Detection** **File: eight.in**

Problem Description

Anyone who knows anything about machine learning has seen the classic example of a program "learning" to recognize to hand-drawn digits. However, such a problem would be too tedious to put in a programming contest, so we have an easier one. Given an array of characters, all either '`.`' or '`#`', which represents a hand-drawn digit, determine whether or not the digit is an eight.

Here is a quick guide to help you on your eight detection journey. A '`.`' on the image is considered a blank spot, and a '`#`' is considered to be a spot with ink. If two spots with ink are horizontally or vertically adjacent, they are considered to be in the same stroke. If two blank spots are horizontally, vertically, or diagonally adjacent, they are considered to be in the same region. If a region of blank space is completely surrounded by ink (i.e. it doesn't contain a border of the image), that region is considered to be a hole. An eight is an image with one stroke that has two holes within it.

Input: The first line begins with a single integer T ($1 \le T \le 50$), denoting the number of images. T digit descriptions follow, each starting with two integers, R and C ($1 \le R, C \le 100$). The next R lines each have C characters and they describe the contents of the image.

Output: For each test case, print a line containing "`Eight!`" if the image is an eight, or "`Not eight.`" if it is not.

Sample Input:

```
6
5 3
###
#.#
###
#.#
###
5 3
##.
#.#
###
#.#
###
```

```
5 3
.#.
#.#
###
#.#
.#.
5 5
###..
#.###
###.#
###.#
.####
```

```
3 8
###..###
#.#..#.#
###..###
7 6
#####.
#..###
###..#
#.#..#
.#####
.#.#..
.###..
```

The sample output is on the next page.

Sample Output:	Eight! Not eight. Not eight. Eight! Not eight. Eight!

Frozen Eagle 2017

Problem 10 **Puya's Mom** **File: mom.in**

Problem Description

Puya's mother has always been an enthusiastic parent. She likes to show her support to her son by attending every event he goes to. Being a professional videographer, she brings her camera with her at all times. Puya greatly appreciates this due to his social media addiction, but does not want his friends to know about it, so he talks to his mom in a secret code. He interlaces the phrase "mom get the camera" within his spoken sentences to let her know when to take pictures, and exactly how many. To be increasingly discrete, he uses individual words from the phrase randomly in normal conversation. The individual words are never within other words (e.g. "There" does not count as the word "The"). His mother, although marginally good at listening, is unfortunately terrible at counting and needs your help to be able to figure out how many pictures her son wants her to take.

Input: The first line begins with a single integer T ($1 \leq T \leq 50$), denoting the number of test cases. Each test case contains a number of sentences terminated by '#'. For simplicity, all sentences will contain only alphanumeric characters and spaces.

Output: For each test case, output the number of possible full "mom get the camera" sentences that can be generated from the input, regardless of order and ignoring capitalization (`Mom` == `mom`).

Sample Input:	2 According to all mom known laws of aviation camera There get is no way the a bee should be able to fly camera Its wings are too small to get its fat little body off the ground The bee of the course flies anyway because bees dont Care what humans think is impossible # I do not think she understands my code anymore She refuses to take pictures I guess I have to be clearer Mom get the camera get the camera Mom please get the camera The camera mom Get the camera now #
Sample Output:	1 3

Frozen Eagle 2017

Problem 11 **Hashing Bashing** **File: hashing.in**

Problem Description

Emmanuel has recently gotten himself into a hacking community (The script kiddies). He's been learning all about how hashes are used in cryptographic scenarios. In fact he is into reversing hashes as well. He found a list of hashed strings, but wants to reverse them. Obviously he knows that hashes aren't reversible, but he would like to know how many possible strings could've given him this hashed value.

The hash is a simple one (also an insecure one) which simply takes the sum of the characters. Each character has a value with '`a`' being 1 and '`z`' being 26. Therefore the hash for the string '`abc`' would be 1 + 2 + 3 = 6. This hash could've also come from '`bca`' or '`bbb`' though. Emmanuel would like to know how many possible strings will give him a given hash value. He will also tell you the length of string to look for. Note that this hash only works on sequences of lower case letters.

Input: The first line begins with a single integer T ($1 \leq T \leq 50$), indicating the number of datasets. Each data set contains two integers L ($1 \leq L \leq 500$) and H ($1 \leq H \leq 13000$) where L is the length of string to look for and H is the given hash value.

Output: For each test case, print a line containing the number of possible strings containing only lowercase letters that could've given this hash value. Output this value modulo 10^9+9.

Sample Input:	`2`
	`3 6`
	`5 5`

Sample Output:	`10` `1`

Frozen Eagle 2017

Problem 12 **Palindromes** **File: palindromes.in**

Problem Description

Syamul Ahmed recently changed his name to Siaam Sarker. Of course Siaam looks easier to pronounce than Syamul even though they are actually pronounced identically (I know right). That's not the most important thing though as Siaam is also closer to a palindrome. Only three characters need to be added to make "siaam" a palindrome while five characters are needed to make "syamul" a palindrome. After facing lots of ridicule due to the odd spelling of his name, he is thinking about changing it again. Being unable to find the minimum number of characters that need to be added for his name to be a palindrome, he needs your help to make his name a palindrome.

Input: The first line begins with a single integer T ($1 \leq T \leq 50$), indicating the number of datasets in the input. Each dataset contains a single line with a string of at least 1 lowercase letter indicating Siaam's new name. The length of this string will be at most 1000 characters.

Output: For each test case, print a line containing the minimum number of characters that must be added to the name to make it a palindrome.

Sample Input:	3 syamul siaam siaaaaam
Sample Output:	5 3 3