Computer Science Competition

2010 Cypress Woods Programming Set

I. General Notes

1. Complete the problems in any order prefered. They do not have to be done in order from 1 to 12.

2. All problems have a value of 60 points. Incorrect submissions receive a 5 point deduction, however may be reworked and resubmitted. Deductions are only included in the team score for problems that are ultimately solved correctly.

3. There is no extraneous input. All input is exactly as specified in the problem. Unless specified by the problem, integer inputs will not have leading zeros. Unless otherwise specified, your program should read to the end of file.

4. Your program should not print extraneous output. Follow the form exactly as given in the problem.

II. Names of Problems

Number	**Name**
Problem 1	Burrows-Wheeler
Problem 2	Characters
Problem 3	Efficiency Calculation
Problem 4	GCD
Problem 5	I Hate Trees
Problem 6	Perimeter
Problem 7	Play Fair
Problem 8	So Easy It's Difficult
Problem 9	Something
Problem 10	Steve Irwin
Problem 11	Transpose It
Problem 12	Welcome to the Nether

01. Burrows-Wheeler

Program Name: Burrowswheeler.java **Input File: burrowswheeler.dat**

You will use the Burrows-Wheeler transformation in order to encrypt and decrypt strings.

A string that needs to be encrypted will have a close character (C.C), `'~'`, at the end of it. Encryption is done by making a rotational identity matrix by rotating the string once for each subsequent row of the matrix. You then sort the matrix by the first letter of each row, and the encrypted string is the last column of the matrix.

Input	All Rotations	Sort the Rows	Output
^BANANA~	^BANANA~	ANANA~^**B**	BNN^AA~A
	~^BANANA	ANA~^BA**N**	
	A~^BANAN	A~^BANA**N**	
	NA~^BANA	BANANA~**^**	
	ANA~^BAN	NANA~^B**A**	
	NANA~^BA	NA~^BAN**A**	
	ANANA~^B	^BANANA**~**	
	BANANA~^	~^BANAN**A**	

Decryption is done by adding each letter of the encoded string at the front of each row in another identity matrix and then sorting the rows. This is repeated until the number of columns is the same as the number of characters in the string. The decrypted string is on the row where the C.C. is the last character.

Add 1	Sort 1	Add 2	Sort 2	Add 3	Sort 3	Add 7	Sort 7	Add 8	Sort 8
B	A	BA	AN	BAN	ANA	BANANA~	ANANA~^	BANANA~^	ANANA~^B
N	A	NA	AN	NAN	ANA	NANA~^B	ANA~^BA	NANA~^BA	ANA~^BAN
N	A	NA	A~	NA~	A~^	NA~^BAN	A~^BANA	NA~^BANA	A~^BANAN
^	B	^B	BA	^BA	BAN	^BANANA	BANANA~	^BANANA~	BANANA~^
A	N	AN	NA	ANA	NAN	ANANA~^	NANA~^B	ANANA~^B	NANA~^BA
A	N	AN	NA	ANA	NA~	ANA~^BA	NA~^BAN	ANA~^BAN	NA~^BANA
~	^	~^	^B	~^B	^BA	~^BANAN	^BANANA	~^BANANA	**^BANANA~**
A	~	A~	~^	A~^	~^B	A~^BANA	~^BANAN	A~^BANAN	~^BANANA

Input

The first line will contain a number, `n`, indicating the number of data sets to follow. On each line there will be two words, the first word `A` will be either `"ENCODED"` or `"DECODED"`, the second word `B` will have a length less than 20 characters long.

Output

For each data set, if `A` is `"DECODED"` then you must output the encoded version of `B`, otherwise you must output the decoded version of `B`. Each output is on a separate line.

Assumptions

If `A` is `"DECODED"`, then `B` will end in `'~'`.

Example Input File

```
3
DECODED ^BANANA~
DECODED ABRAKADABRA~
ENCODED ENLPPI~PAE
```

Example Output to Screen

```
BNN^AA~A
~DKRRAAAABBA
PINEAPPLE~
```

02. Characters

Program Name: Characters.java **Input File: characters.dat**

In order to fool the teachers in this world who take and read our notes without our consent, we've discovered a way of making them incomprehensible by adding a non-related character to most of our words that have the same lengths. Your goal will be to take in a list of words and determine the non-related character that, when removed, would make all of the words the same length.

Input

The first line will contain a number, indicating the number of data sets to follow. The following data sets will consist of 5 strings, each on its own line. Data sets will NOT be separated by a blank line. You must search which character, when removed from all the strings, will result in 5 strings of equal length.

Output

The first line of output will contain the character that, when removed, results in 5 strings of equal length by printing `"REMOVE"`, followed by the character. You will then print, each on their own line, the 5 strings with the character removed.

Assumptions

All sets will have a solution. Final strings will not be longer than 26 characters. However, input strings may be longer.

Example Input File

```
2
kaaiaoaaealasac
awcaeaoeanag
aiangailabe
tyeaaaaaaaaaaantayo
cronicla
ruffles
fiffalenfff
roffsefl
fanfare
nilfen
```

Example Output to Screen

```
REMOVE a
kioelsc
wceoeng
ingilbe
tyentyo
cronicl
REMOVE f
rules
ialen
rosel
anare
nilen
```

03. Efficiency Calculation

Program Name: Eff.java **Input File: eff.dat**

Bob the Builder has been contracted to rebuild the Leaning Tower of Pisa in order to make a significantly more stable structure. Bob built several miniature trial towers to test for structural flaws and has recorded their masses and weight capacities. Efficiency is determined using this formula: E = W/M g., where E is the efficiency, W is the weight capacity in grams, and M is the mass of the tower in grams.

Input

The first line will contain a number, `n,` that represents the number of data sets to follow. There can be up to 100 data sets, each containing two positive integers less than `Integer.MAX_VALUE` and greater than `Integer.MIN_VALUE.` The first integer, `w,` will be the weight capacity in grams and the second integer, `m,` will be the mass of the tower in grams.

Output

Output the efficiency of Bob's tower using the supplied formula. Disregard any decimal places and output an integer.

Example Input File

```
3
15432 13
13382 7
500 56
```

Example Output to Screen

```
1187
1911
8
```

04. GCD

Program Name: GCD.java   Input File: GCD.dat

Recently during an intense recreational football game, Nik gave Luke a traumatizing concussion. As a result, Luke has forgotten how to perform many basic functions in life, such as playing Minecraft™. Luke's Algebra II teacher now demands that Luke do his homework which consists primarily of problems involving the greatest common denominator. However, due to his fatal head injury, he is unable to complete this daunting assignment. Help Luke write a correctly working program that does his GCD homework.

Input

The first line will contain number a and the second line will contain number b, where $0 < a \le b < 2^{12}$.

Output

Print out the greatest common factor / denominator of a and b.

Example Input File

```
12
16
```

Example Output to Screen

```
4
```

05. I Hate Trees

Program Name: Deforestation.java **Input File: deforestation.dat**

Nik goes mad after failing a spelling test. As a result, he decides to take his trusty diamond axe and rages through the Amazon, chopping down every tree that catches his malignant eye. Luckily, the police have recently locked him up, and now it is your job to restore the beautiful forests of the world with the most complex and resilient trees mankind has ever seen. Good luck...

Input

The first line of input will contain a single integer, i, indicating the number of data sets to process. The remainder of the input consists of those n data sets.

Each data set will consist of a list numbers no longer than 2^{12} and no shorter than 1. Each individual number will be no greater than ∞ and no less than 0

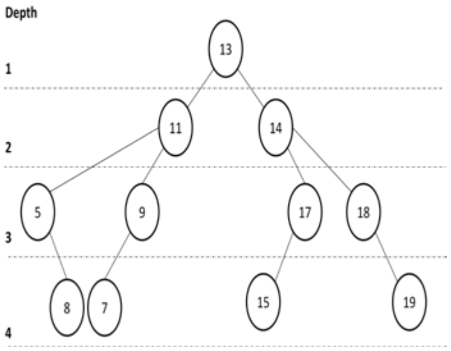
Output

The list of data must be put into a complex binary tree. Every reference to depth will consider the root has a depth of one. Each node may have a maximum of left and right children as it is deep. For example the child of the root may have two left and two right children.

If a node has not been filled to its maximum of left children and the next number is less than or equal to the node and its farthest left child then it should be added as a child to the current node. If the next number falls between two of the left children then add the number to the child with the least number of children that apply. For example if it falls between a child's left nodes with values 5 and 9 (9 being closer to the child) only 9's left nodes apply and 5's right nodes because the node would be add to 9's left or 5's right. In the case of a tie add the next number to the child closest to the node. If it falls between the node and a child add the number to the child. All these rules apply for the right nodes in reverse. (This is not a trick just change the signs and small stuff like that). Other than this it acts like a standard binary tree.

Standard Complex Tree

13 14 17 18 11 19 15 9 5 7 8



This data of the Tree implements a specific comparable. Prime numbers a bigger than composite. Even numbers are bigger than odd numbers. Then it compares by value.

Output a modified post order traversal. (Leftmost – Rightmost children, Node)

When printing the node output it in the form of `"baseX(n)"` where X is the number of children the node has and n is the value of the node in base X. If the number of children is 0 or 1 X will then become 10.

If multiple number of the same base are to be outputted consecutively output them as `"baseX(n,n1,n2,…)"`

Example Input File

```
4
4 7 6 5
7 3
3 1 7 9 2
9 1 5 7 8 6 3 2 1 4 8
```

Example Output to Screen

```
base10(5,6,7,4)
base10(3,7)
base10(9,1,2,7) base2(11)
base10(1,1,4,6,8,3) base2(1000) base10(7,2) base4(11) base2(1001)
```

06. Perimeter

Program Name: Perimeter.java **Input File: perimeter.dat**

Mr. and Mrs. Smith want to start expanding financially by entering real estate in order to accumulate the funds needed to start a toothbrush company. When deciding which houses to purchase for the purpose of leasing, location is of course the main priority in making these important decisions. However, arguably of course, the next most important factor to consider when purchasing a house is the features that coincide with the house. A pool, being one of the most imperative influences, is a considerable parameter that both Smiths took into consideration. Thus, the Smiths decide that the most strategic way to accumulate leasers is to purchase a house in the right location with a pool, and with all the pools coincidentally being in the shape of perfect circles.

In order to correctly advertise their homes, the Smiths have to provide accurate measurements of both the volume and perimeter of their pools, according to regulations. Being English majors, both the Smiths have forgotten how to calculate the perimeters of their pools, but have luckily remembered how to correctly measure the diameter of their pools. Your goal is to help the Smiths calculate the perimeters of their pools so that they can get some money to start a toothbrush company!

Input

The first number signifies the number of diameters to follow, each of which will be less than `100000.`

Output

Print out the perimeters of the circles to 5 decimal places on separate lines.

Example Input File

```
3
14
82
76478
```

Example Output to Screen

```
43.98230
257.61060
240262.72296
```

07. Play Fair

Program Name: Playfair.java　　　　**Input File: playfair.dat**

The Playfair cipher uses a 5 by 5 table containing a unique keyword. You will be given a pre-generated 5 by 5 table which will be used in the encryption process. To encrypt a message, one would first break a given message into digraphs (groups of 2 letters), replacing all `'J'`s with `'I'`s. The letters of each digraph must be different; therefore, an `'X'` is added after the first letter of each digraph that contains duplicates. An `'X'` is appended to the last digraph if it only contains one letter.

Observe the following example, given the message, “Hide the gold in the tree stump”. As seen, an `'X'` is inserted between the digraphs containing duplicate `'E'`s.

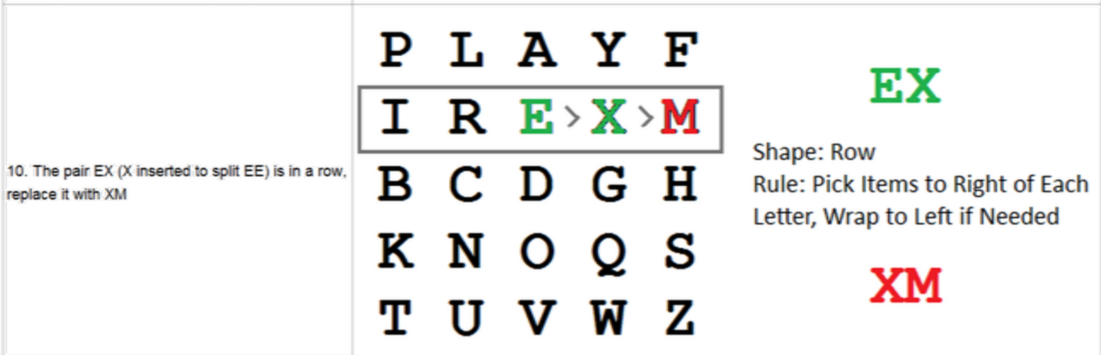
```
HI DE TH EG OL DI NT HE TR EE ST UM P

HI DE TH EG OL DI NT HE TR EX ES TU MP
                            ^
```

There will not be 2 `'X's` in a row, ever. (ever ever)

Once the digraphs have been formed, one should apply the following rules to each digraph, given the following example table:

- If the letters appear on the same row of your table, replace them with the letters to their immediate right respectively (wrapping around to the left side of the row if a letter in the original pair was on the right side of the row).



10. The pair EX (X inserted to split EE) is in a row, replace it with XM

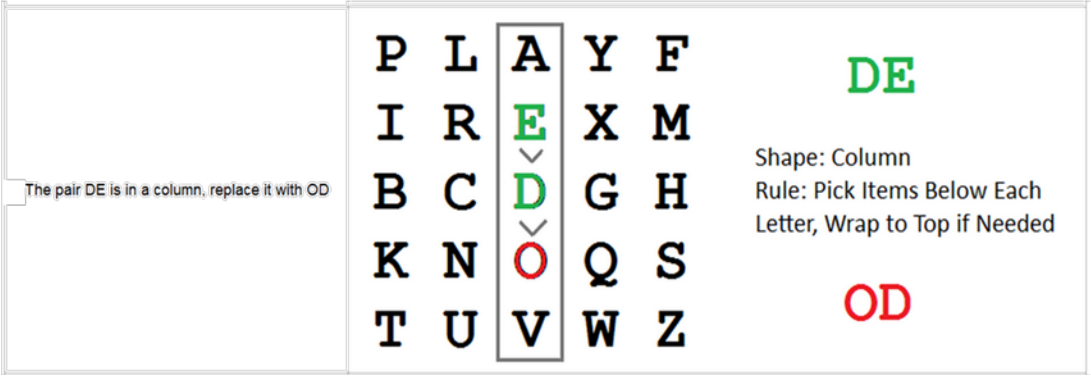
P	L	A	Y	F
I	R	E	X	M
B	C	D	G	H
K	N	O	Q	S
T	U	V	W	Z

EX

Shape: Row
Rule: Pick Items to Right of Each Letter, Wrap to Left if Needed

XM

- If the letters appear on the same column of your table, replace them with the letters immediately below respectively (wrapping around to the top side of the column if a letter in the original pair was on the bottom side of the column).



The pair DE is in a column, replace it with OD

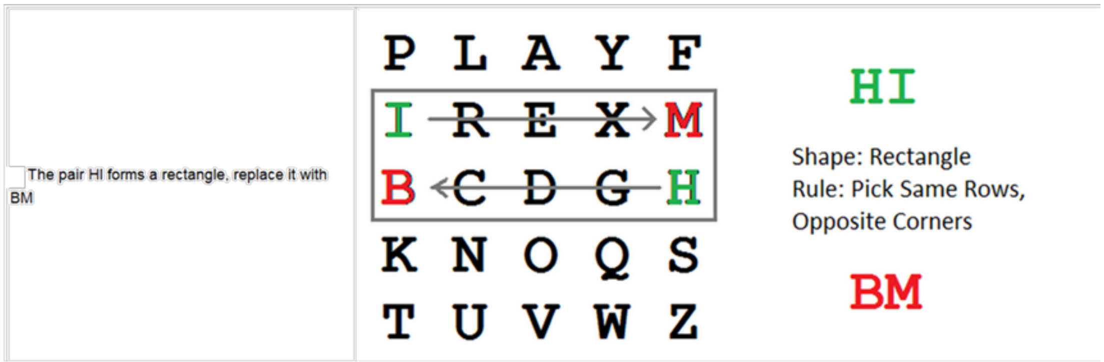
P	L	A	Y	F
I	R	E	X	M
B	C	D	G	H
K	N	O	Q	S
T	U	V	W	Z

DE

Shape: Column
Rule: Pick Items Below Each Letter, Wrap to Top if Needed

OD

- If the letters are not on the same row or column, replace them with the letters on the same row respectively but at the other pair of corners of the rectangle defined by the original pair. The order is important – the first letter of the encrypted pair is the one that lies on the same row as the first letter of the plaintext pair.



The pair HI forms a rectangle, replace it with BM

When all encrypted diagraphs are combined without spaces, the message, "Hide the gold in the tree stump", becomes `"BMODZBXDNABEKUDMUIXMMOUVIF".`

Input

The first five lines contain a 5 by 5 table which will be used to encrypt the digraphs. The next line after that contains an integer, `n`, that indicates the amount of messages to follow. All messages will be encrypted using the same given table. All messages to be encrypted may or may not contain spaces, punctuation, and capitalization. They should be broken into digraphs and encrypted using the given table during the encryption process.

Output

For each data set, your task is to output the encrypted message with all spaces and punctuation removed and in all uppercase.

Example Input File

```
PLAYF
IREXM
BCDGH
KNOQS
TUVWZ
3
Hide the gold in the tree stump
Computer Science is awesome!
I love Playfair!
```

Example Output to Screen

```
BMODZBXDNABEKUDMUIXMMOUVIF
DNIFVUXENHRXUNXROFVXKQIX
RPVAIAAYFPPEEM
```

08. So Easy It’s Difficult

Program Name: Seid.java **Input File: none**

Sugar, spice, and everything nice
These were the ingredients chosen
To create the dinosaur named Stephen
But Professor Utonium accidentally
Added an extra ingredient to the concoction–
Chemical X
Thus, The Stephen was born
Using his ultra-super powers
Stephen
Has dedicated his life to fighting crime
And the forces of evil.
Using your extensive knowledge of computer science and java carefully print the following output.

Input

NONE

Output

Print out the following dinosaur

Example Output to Screen

```
               __
              /*_)
     _.----._/ /
    /         /
 __/ (  | (  |
/__.- |_|--|_|
```

09. Something

Program Name: Something.java **Input File: something.dat**

When deleting a virus, a computer scientist genius decides that the only way to delete a Trojan horse is to delete every single file that the virus has infected, and every file that precedes and follows it. For manageable means, the antivirus software has decided to name every single file with numbers and assign a specific number to the virus. Help the computer science genius come up with an algorithm that corresponds to his solution to this disastrous virus.

Input

The first line of input will contain a single integer, n, indicating the number of data sets to process. The remainder of the input consists of those n data sets.

Each data set will consist of two parts

The first part of the input will consist of a list of numbers separated by one space followed by `~` `(the numbers in the list will not exceed 2`^{33} `or be less than -2`^{33}`)`

Directly after the `~` will be a number `n (n will not exceed 2`^{33} `or be less than -2`^{33}`)`

Output

Find the mathematical mean of the list disregarding the number before and after every instance of the specified value `(n)`. Disregard the number as well.

If there is no number before the number (or after) do not wrap around.

Example Input File

```
3
1 2 3 4 5 3 5 ~ 3
1 3 47 89 0 -48 -4 7849 492 12 ~ 12
0 1 0 0 0 0 0 8 0 1 1 1 1 1 1 1 1 1 8589934591 ~ 8
```

Example Output to Screen

```
1
992
536870912
```

10. Steve Irwin

Program Name: SteveIrwin.java **Input File: steveirwin.dat**

When going through the belongings of the late Steve Irwin (The Crocodile Hunter) to divide up his estate, court officials discovered a booklet in which Steve had compiled the most comprehensive guide and informational bible of reptilian and amphibian knowledge known to man. He was a bit messy and wrote entries for each species he encountered in a kind of shorthand. Your duty is to analyze Steve's data and print out all of the reptiles and amphibians ordered by their dangerous-ness. Dangerous-ness will be determined by the sum of all traits values.

Steve noted colors and patterns on the reptiles and this table will need to be used to analyze the traits of the reptile or amphibian. Different traits mean different things for different reptiles or amphibians.

Trait	Snakes	Frogs	Lizards
Yellow	+1	+2	-1
Red	+1	+4	+3
White	+5	-2	-2
Spotted	-2	+4	-4
Striped	+6	-2	-5

Input
The first line will contain an integer denoting how many entries are in the diary. Each entry will be on its own line and consists of a name and, separated by a `"@"`, one contiguous sequence. Yellow is denoted by `"Y",` Red by `"R",` White by `"W",` Spotted by `".",` and Striped by `"-"`. The type of reptile or amphibian will be indicated by the first letter of the string or traits. An `"F"` means Frog, an `"S"` means Snake, and an `"L"` means Lizard.

Output
For each data set, output a list of the reptiles in order. Dangerousness is determined one digit at a time. Print first the reptiles or amphibians score and then its name separated by a single space. In the event than multiple entries have the same score, sort by its name.

Example Input File
```
4
Shiny Frog@F-WY
Spotted Snake@SR.Y
Comp Sci Lizard@LY.RW-
Speckled Psycho Killer Frog@FY.R
```

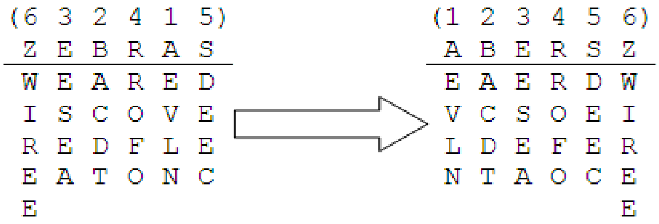
Example Output to Screen
```
-2 Shiny Frog
-9 Comp Sci Lizard
0 Spotted Snake
10 Speckled Psycho Killer Frog
```

11. Transpose It

Program Name: Transpose.java **Input File: transpose.dat**

In a columnar transposition cipher, a given message is written out in a grid of rows of a fixed length, and then read off again in columns; the order of these columns are determined by the alphabetical order of the letters of a given keyword.

For example, given the keyword "ZEBRAS" of length 6, the rows are of length 6, and the order by which one reads the columns is defined by the alphabetical order of the letters in the keyword. In this case, the order would be "632415". When filling out the grid, one fills it with the letters of the message in order from left to right. For example, using the keyword “ZEBRAS” and the message "WE ARE DISCOVERED. FLEE AT ONCE", the grid will be filled out thus, with spaces between words removed and any extra spots left empty:



```
(6 3 2 4 1 5)      (1 2 3 4 5 6)
 Z E B R A S        A B E R S Z
 W E A R E D        E A E R D W
 I S C O V E        V C S O E I
 R E D F L E        L D E F E R
 E A T O N C        N T A O C E
 E                            E
```

The ciphertext is then read off in the order of the numbered columns to form one body of text which is output in groups of five. This example results in the encrypted text: “EVLNA CDTES EAROF ODEEC WIREE”.

Input

The first line of input will contain a single integer, *n*, indicating the number of data sets to follow. Each data set will be composed of two lines:

- The first line contains a unique keyword that does not contain duplicate letters, is capitalized throughout and has a length is less than that of the message.
- The second line contains the message to be encrypted, which may or may not contain punctuation and capitalization.

Output

For each data set, your task is to output the encrypted message in groups of five separated by spaces, containing no punctuation and in all uppercase. The last group output does not necessarily contain five letters.

Assumptions

There will be no keyword with duplicate letters.

Example Input File

```
3
TRANS
Position cipher!
COMPILE
CoMp ScI iS tOo AwEsOmE
ABZDJ
Let us destroy the enemy.
```

Example Output to Screen

```
SNEIC ROOHT IPIP
CIEIW SOECA MTOOS SPOM
LDOEE EYNUT HMSRE YTSTE
```

12. Welcome To The Nether

Program Name: Helpme.java Input File: helpme.dat

One day while Henry was digging to the center of the earth on Minecraft ™, he encountered a pit of lava. However, this lava is somewhat special. It tears the dimensional rift of the universe, resulting in a skewed plane of existence in the area that the lava encompasses. This creates a four dimensional area as a result of the daunting portal made of obsidian hidden in the lava. Consequently, Henry's compass that he created out of redstone and iron ingots is malfunctioning. Write a program to help Henry find his way home to the blocky world of Minecraft ™, lest he die a fiery death, get exploded by creepers, and lose all of his diamond tools.

Input

The first line of input will contain a single integer, n, indicating the number of data sets to process. The remainder of the input consists of those n data sets.

Each data set will consist of two parts

1. There will then be 4 integer `"h ih w iw"`, where `w` (2-40) is the width of the maze, `h` (2-40) is the height of the maze, ih (2-40) is the height of the inner maze, and `iw` (2-40) is the width of the inner maze. (40x40x40x40 is very large, think efficiency)

2. The next h sets of ih lines will each contain w sets of iw characters, (separated by spaces) with each character being one of the following:
   - `The most appearing letter`—This represents a possible route to the exit
   - `'A'`—This represents the start
   - `'Z'`—This represents the exit
   - `Anything not a route start or exit`—This represents lava

Output

Mazes like this move in a special way. You can move to adjacent spots within the inner maze you're currently in as well as move to the same spot to adjacent mazes. You cannot move diagonally in any way.

Output the number of steps in the format of `"Solved in x at spot [h,ih,w,iw]"` where `x` is the number of steps

Assumptions

There will always be an exit. There will always be one letter that appears most often. There will only be CAPITAL letters in the maze.

Example Input File

```
2
2 5 4 2
A T  P P  I R  X L
Y W  F P  G M  I P
H K  V P  L Y  K D
I N  P P  Q U  I F
F S  P G  P X  Q O

M I  B G  P H  R X
C P  T X  P X  Q O
W E  O G  P F  Q I
D H  I G  P G  J I
W H  M C  P V  P Z


6 2 2 6
A P P E D U  V T L P L M
V K P X I Q  O N F U I D

B H C U T M  L W X T D G
G S P W I P  W C R F K M

D O P P G L  D F P Y J U
Q B P T E U  U I K R L L

R W W W F M  I U P P P P
S H M D D W  O Q M Q X D

L C U Y B F  Q O W M M P
F S K E K H  E Y K X T E

O G K K W N  N D G H B P
J S H R K R  T R P D C Z
```

Example Output to Screen

```
Solved in 11 at spot [1,4,3,1]
Solved in 14 at spot [5,1,1,5]
```