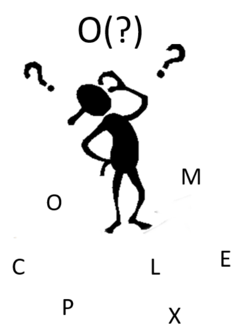
Pacific Northwest Regional Programming Contest
Systems Test and Practice Contest

27 February 2021

- Some problems are duplicated in this practice contest. This is to enable additional testing of the system.
- The languages supported are C, C++ 17 (with Gnu extensions), Java 11, Python 3 (with pypy3), and Kotlin.
- The name of the file you submit must not have spaces in it.
- Python 2 and C# are not supported this year.
- For all problems, read the input data from standard input and write the results to standard output.
- In general, when there is more than one integer or word on an input line, they will be separated from each other by exactly one space. No input lines will have leading or trailing spaces, and tabs will never appear in any input.
- Python may not have sufficient performance for many of the problems; use it at your discretion.

Problem N
Complexity1
Time Limit: 2



Define the *complexity* of a string to be the number of distinct letters in it. For example, the string `string` has complexity 6 and the string `letter` has complexity 4.

You like strings which have complexity either 1 or 2. Your friend has given you a string and you want to turn it into a string that you like. You have a magic eraser which will delete one letter from any string. Compute the minimum number of times you will need to use the eraser to turn the string into a string with complexity at most 2.

Input

The input consists of a single line that contains a single string of at most 100 lowercase ASCII letters (‘`a`’–‘`z`’).

Output

Print, on a single line, the minimum number of times you need to use the eraser.

Sample Input 1	Sample Output 1
`string`	`4`

Sample Input 2	Sample Output 2
`letter`	`2`

Sample Input 3	Sample Output 3
`aaaaaa`	`0`

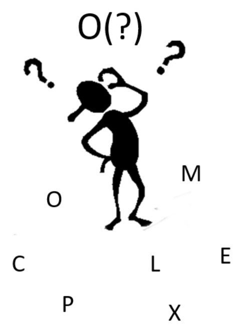
Sample Input 4	Sample Output 4
`uncopyrightable`	`13`

Sample Input 5	Sample Output 5
`ambidextrously`	`12`

Sample Input 6	Sample Output 6
`assesses`	`1`

Sample Input 7	Sample Output 7
`assassins`	`2`

Problem O
Complexity2
Time Limit: 2



Define the *complexity* of a string to be the number of distinct letters in it. For example, the string `string` has complexity 6 and the string `letter` has complexity 4.

You like strings which have complexity either 1 or 2. Your friend has given you a string and you want to turn it into a string that you like. You have a magic eraser which will delete one letter from any string. Compute the minimum number of times you will need to use the eraser to turn the string into a string with complexity at most 2.

Input

The input consists of a single line that contains a single string of at most 100 lowercase ASCII letters (‘`a`’–‘`z`’).

Output

Print, on a single line, the minimum number of times you need to use the eraser.

Sample Input 1	Sample Output 1
`string`	`4`

Sample Input 2	Sample Output 2
`letter`	`2`

Sample Input 3	Sample Output 3
`aaaaaa`	`0`

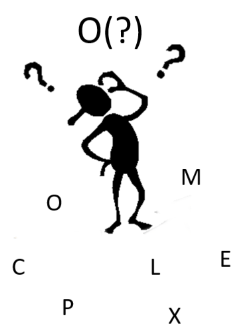
Sample Input 4	Sample Output 4
`uncopyrightable`	`13`

Sample Input 5	Sample Output 5
`ambidextrously`	`12`

Sample Input 6	Sample Output 6
`assesses`	`1`

Sample Input 7	Sample Output 7
`assassins`	`2`

Problem P
Complexity3
Time Limit: 2



Define the *complexity* of a string to be the number of distinct letters in it. For example, the string `string` has complexity 6 and the string `letter` has complexity 4.

You like strings which have complexity either 1 or 2. Your friend has given you a string and you want to turn it into a string that you like. You have a magic eraser which will delete one letter from any string. Compute the minimum number of times you will need to use the eraser to turn the string into a string with complexity at most 2.

Input

The input consists of a single line that contains a single string of at most 100 lowercase ASCII letters (‘`a`’–‘`z`’).

Output

Print, on a single line, the minimum number of times you need to use the eraser.

Sample Input 1	Sample Output 1
`string`	`4`

Sample Input 2	Sample Output 2
`letter`	`2`

Sample Input 3	Sample Output 3
`aaaaaa`	`0`

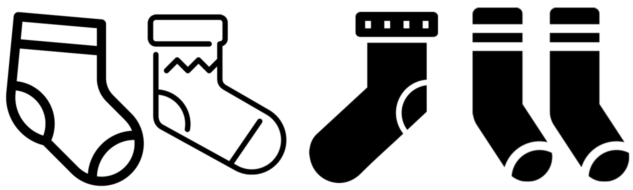
Sample Input 4	Sample Output 4
`uncopyrightable`	`13`

Sample Input 5	Sample Output 5
`ambidextrously`	`12`

Sample Input 6	Sample Output 6
`assesses`	`1`

Sample Input 7	Sample Output 7
`assassins`	`2`

Problem Q
MismatchedSocks1
Time Limit: 1



Fred likes to wear mismatched socks. This sometimes means he has to plan ahead.

Suppose his sock drawer has 1 red, 1 blue, and 2 green socks. If he wears the red with the blue, he is stuck with matching green socks the next day. Given the contents of his sock drawer, how many pairs of mismatched socks can he put together?

Input

The first line of input contains a single integer n ($1 \leq n \leq 1{,}000$), the number of different colors of socks in Fred's drawer.

The ith of the next n lines contains a single integer k_i ($1 \leq k_i \leq 10^9$), the number of socks of the ith color.

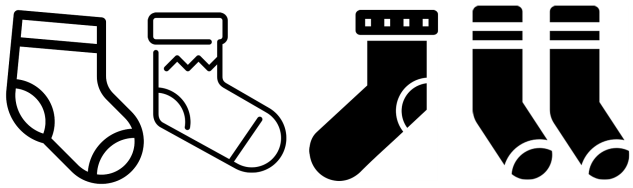
Output

Print, on a single line, the maximum number of mismatched pairs of socks that Fred can make with the contents of his sock drawer.

Sample Input 1	Sample Output 1
3 1 2 1	2

Sample Input 2	Sample Output 2
5 1 2 1 10 3	7

Problem R
MismatchedSocks2
Time Limit: 1



Fred likes to wear mismatched socks. This sometimes means he has to plan ahead.

Suppose his sock drawer has 1 red, 1 blue, and 2 green socks. If he wears the red with the blue, he is stuck with matching green socks the next day. Given the contents of his sock drawer, how many pairs of mismatched socks can he put together?

Input

The first line of input contains a single integer n ($1 \leq n \leq 1{,}000$), the number of different colors of socks in Fred's drawer.

The ith of the next n lines contains a single integer k_i ($1 \leq k_i \leq 10^9$), the number of socks of the ith color.

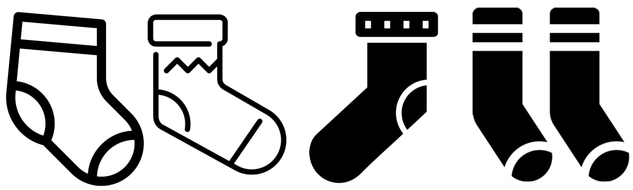
Output

Print, on a single line, the maximum number of mismatched pairs of socks that Fred can make with the contents of his sock drawer.

Sample Input 1	Sample Output 1
3 1 2 1	2

Sample Input 2	Sample Output 2
5 1 2 1 10 3	7

Problem S
MismatchedSocks3
Time Limit: 1



Fred likes to wear mismatched socks. This sometimes means he has to plan ahead.

Suppose his sock drawer has 1 red, 1 blue, and 2 green socks. If he wears the red with the blue, he is stuck with matching green socks the next day. Given the contents of his sock drawer, how many pairs of mismatched socks can he put together?

Input

The first line of input contains a single integer n ($1 \leq n \leq 1{,}000$), the number of different colors of socks in Fred's drawer.

The ith of the next n lines contains a single integer k_i ($1 \leq k_i \leq 10^9$), the number of socks of the ith color.

Output

Print, on a single line, the maximum number of mismatched pairs of socks that Fred can make with the contents of his sock drawer.

Sample Input 1	Sample Output 1
3 1 2 1	2

Sample Input 2	Sample Output 2
5 1 2 1 10 3	7

Problem T
Tongues1
Time Limit: 5

Gandalf's writings have long been available for study, but no one has yet figured out what language they are written in. Recently, due to programming work by a hacker known only by the code name ROT13, it has been discovered that Gandalf used nothing but a simple letter substitution scheme, and further, that it is its own inverse—the same operation scrambles the message as unscrambles it.

This operation is performed by replacing vowels in the sequence

> (a i y e o u)

with the vowel three advanced, cyclicly, while preserving case (i.e., lower or upper). Similarly, consonants are replaced from the sequence

> (b k x z n h d c w g p v j q t s r l m f)

by advancing ten letters. So for instance the phrase

> One ring to rule them all.

translates to

> Ita dotf ni dyca nsaw ecc.

The fascinating thing about this transformation is that the resulting language yields pronounceable words.

For this problem, you will write code to translate Gandalf's manuscripts into plain text.

Input

The input file will contain multiple test cases. Each test case consists of a single line containing up to 100 characters, representing some text written by Gandalf. All characters will be plain ASCII, in the range space (32) to tilde (126), plus a newline terminating each line. The end of the input is denoted by the end-of-file.

Output

For each input test case, print its translation into plaintext. The output should contain exactly the same number of lines and characters as the input.

Sample Input 1	Sample Output 1
`Ita dotf ni dyca nsaw ecc.`	`One ring to rule them all.`

Problem U
Tongues2
Time Limit: 5

Gandalf's writings have long been available for study, but no one has yet figured out what language they are written in. Recently, due to programming work by a hacker known only by the code name ROT13, it has been discovered that Gandalf used nothing but a simple letter substitution scheme, and further, that it is its own inverse—the same operation scrambles the message as unscrambles it.

This operation is performed by replacing vowels in the sequence

(a i y e o u)

with the vowel three advanced, cyclicly, while preserving case (i.e., lower or upper). Similarly, consonants are replaced from the sequence

(b k x z n h d c w g p v j q t s r l m f)

by advancing ten letters. So for instance the phrase

One ring to rule them all.

translates to

Ita dotf ni dyca nsaw ecc.

The fascinating thing about this transformation is that the resulting language yields pronounceable words.

For this problem, you will write code to translate Gandalf's manuscripts into plain text.

Input

The input file will contain multiple test cases. Each test case consists of a single line containing up to 100 characters, representing some text written by Gandalf. All characters will be plain ASCII, in the range space (32) to tilde (126), plus a newline terminating each line. The end of the input is denoted by the end-of-file.

Output

For each input test case, print its translation into plaintext. The output should contain exactly the same number of lines and characters as the input.

Sample Input 1	Sample Output 1
`Ita dotf ni dyca nsaw ecc.`	`One ring to rule them all.`

Problem V
Tongues3
Time Limit: 5

Gandalf's writings have long been available for study, but no one has yet figured out what language they are written in. Recently, due to programming work by a hacker known only by the code name ROT13, it has been discovered that Gandalf used nothing but a simple letter substitution scheme, and further, that it is its own inverse—the same operation scrambles the message as unscrambles it.

This operation is performed by replacing vowels in the sequence

(a i y e o u)

with the vowel three advanced, cyclicly, while preserving case (i.e., lower or upper). Similarly, consonants are replaced from the sequence

(b k x z n h d c w g p v j q t s r l m f)

by advancing ten letters. So for instance the phrase

One ring to rule them all.

translates to

Ita dotf ni dyca nsaw ecc.

The fascinating thing about this transformation is that the resulting language yields pronounceable words.

For this problem, you will write code to translate Gandalf's manuscripts into plain text.

Input

The input file will contain multiple test cases. Each test case consists of a single line containing up to 100 characters, representing some text written by Gandalf. All characters will be plain ASCII, in the range space (32) to tilde (126), plus a newline terminating each line. The end of the input is denoted by the end-of-file.

Output

For each input test case, print its translation into plaintext. The output should contain exactly the same number of lines and characters as the input.

Sample Input 1	Sample Output 1
`Ita dotf ni dyca nsaw ecc.`	`One ring to rule them all.`

Problem W
Bones
Time Limit: 5

Bones is investigating what electric shuttle is appropriate for his mom's school district vehicle. Each school has a charging station. It is important that a trip from one school to any other be completed with no more than K rechargings. The car is initially at zero battery and must always be recharged at the start of each trip; this counts as one of the K rechargings. There is at most one road between each pair of schools, and there is at least one path of roads connecting each pair of schools. Given the layout of these roads and K, compute the necessary range required of the electric shuttle.

Input

Input begins with a line with one integer T ($1 \leq T \leq 50$) denoting the number of test cases. Each test case begins with a line containing three integers N, K, and M ($2 \leq N \leq 100$, $1 \leq K \leq 100$), where N denotes the number of schools, K denotes the maximum number of rechargings permitted per trip, and M denotes the number of roads. Next follow M lines each with three integers u_i, v_i, and d_i ($0 \leq u_i, v_i < N$, $u_i \neq v_i$, $1 \leq d_i \leq 10^9$) indicating that road i connects schools u_i and v_i (0-indexed) bidirectionally with distance d_i.

Output

For each test case, output one line containing the minimum range required.

Sample Input 1	Sample Output 1
2 4 2 4 0 1 100 1 2 200 2 3 300 3 0 400 10 2 15 0 1 113 1 2 314 2 3 271 3 4 141 4 0 173 5 7 235 7 9 979 9 6 402 6 8 431 8 5 462 0 5 411 1 6 855 2 7 921 3 8 355 4 9 113	300 688

Problem X
Crusher
Time Limit: 20

Wesley Crusher is the teaching assistant for Introduction to Algorithms. During his first class, the cadets were asked to come up with their own sorting algorithms. Monty came up with the following code:

```
while (!sorted(a)) {
   int i = random(n) ;
   int j = random(n) ;
   if (a[min(i,j)] > a[max(i,j)])
      swap(a[i], a[j]) ;
}
```

Carlos, inspired, came up with the following code:

```
while (!sorted(a)) {
   int i = random(n-1) ;
   int j = i + 1 ;
   if (a[i] > a[j])
      swap(a[i], a[j]) ;
}
```

Wesley needs to determine which algorithm is better.

For a given input array of up to 8 values, calculate and print the expected number of iterations for each algorithm. That is, on average, how many iterations should each algorithm take for the given input?

Input

The first line contains T, the number of test cases: $2 \leq T \leq 100$.

Each test case is given on a single line. The first value is N, the number of array elements; $2 \leq N \leq 8$. This is followed on the same line by N integer array elements. The array elements will have values between 0 and 100 inclusive. The array elements may not be distinct.

Output

For each test case, print out the expected number of iterations for Monty's algorithm and for Carlos's algorithm, as shown in the sample output section. There should be exactly one space between words and no spaces at the start of each line or at the end of each line. There should be exactly six digits after the decimal point. Rounding should be to nearest representable value.

Sample Input 1	Sample Output 1
12 2 1 2 2 2 1 3 1 2 3 3 3 2 1 4 1 2 3 4 4 4 3 2 1 4 2 1 4 3 5 1 1 1 1 1 5 5 4 3 2 1 8 8 7 6 5 4 3 2 1 8 3 1 4 1 5 9 2 6 8 2 7 1 8 2 8 1 8	Monty 0.000000 Carlos 0.000000 Monty 2.000000 Carlos 1.000000 Monty 0.000000 Carlos 0.000000 Monty 6.000000 Carlos 5.000000 Monty 0.000000 Carlos 0.000000 Monty 14.666667 Carlos 12.500000 Monty 12.000000 Carlos 4.500000 Monty 0.000000 Carlos 0.000000 Monty 26.382275 Carlos 23.641975 Monty 89.576273 Carlos 79.496510 Monty 79.161905 Carlos 33.422840 Monty 63.815873 Carlos 38.910494

Problem Y
Escape
Time Limit: 20

The Enterprise is surrounded by Klingons! Find the escape route that has the quickest exit time, and print that time.

Input is a rectangular grid; each grid square either has the Enterprise or some class of a Klingon warship. Associated with each class of Klingon warship is a time that it takes for the Enterprise to defeat that Klingon. To escape, the Enterprise must defeat each Klingon on some path to the perimeter. Squares are connected by their edges, not by corners (thus, four neighbors).

Input

The first line will contain T, the number of cases; $2 \leq T \leq 100$. Each case will start with line containing three numbers k, w, and h. The value for k is the number of different Klingon classes and will be between 1 and 25, inclusive. The value for w is the width of the grid and will be between 1 and 1000, inclusive. The value for h is the height of the grid and will be between 1 and 1000, inclusive.

Following that will be k lines. Each will consist of a capital letter used to label the class of Klingon ships followed by the duration required to defeat that class of Klingon. The label will not be "`E`". The duration is in minutes and will be between 0 and 100,000, inclusive. Each label will be distinct.

Following that will be h lines. Each will consist of w capital letters (with no spaces between them). There will be exactly one "`E`" across all h lines, denoting the location of the Enterprise; all other capital letters will be one of the k labels given above, denoting the class of Klingon warship in the square.

Output

Your output should be a single integer value indicating the time required for the Enterprise to escape.

Sample Input 1	Sample Output 1
2 6 3 3 A 1 B 2 C 3 D 4 F 5 G 6 ABC FEC DBG 2 6 3 A 100 B 1000 BBBBBB AAAAEB BBBBBB	2 400

Problem Z
Good versus Evil
Time Limit: 5

Middle Earth is about to go to war. The forces of good will have many battles with the forces of evil. Different races will certainly be involved. Each race has a certain 'worth' when battling against others. On the side of good we have the following races, with their associated worth:

Hobbits - 1
Men - 2
Elves - 3
Dwarves - 3
Eagles - 4
Wizards - 10

On the side of evil we have:

Orcs - 1
Men - 2
Wargs - 2
Goblins - 2
Uruk Hai - 3
Trolls - 5
Wizards - 11

Although weather, location, supplies and valor play a part in any battle, if you add up the worth of the side of good and compare it with the worth of the side of evil, the side with the larger worth will tend to win.

Thus, given the count of each of the races on the side of good, followed by the count of each of the races on the side of evil, determine which side wins.

Input

The first line of input will contain an integer greater than 0 signifying the number of battles to process. Information for each battle will consist of two lines of data as follows.

First, there will be a line containing the count of each race on the side of good. Each entry will be separated by a single space. The values will be ordered as follows: Hobbits, Men, Elves, Dwarves, Eagles, Wizards.

The next line will contain the count of each race on the side of evil in the following order: Orcs, Men, Wargs, Goblins, Uruk Hai, Trolls, Wizards.

All values are non-negative integers. The resulting sum of the worth for each side will not exceed the limit of a 32-bit integer.

Output

For each battle, print "`Battle`" followed by a single space, followed by the battle number starting at 1, followed by a "`:`", followed by a single space. Then print "`Good triumphs over Evil`" if good wins. Print "`Evil eradicates all trace of Good`" if evil wins. If there is a tie, then print "`No victor on this battle field`".

Sample Input 1	Sample Output 1
3 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 10 0 1 1 1 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0	Battle 1: Evil eradicates all trace of Good Battle 2: Good triumphs over Evil Battle 3: No victor on this battle field

Problem AA
Number Game
Time Limit: 5

Alice and Bob are playing a game on a line of N squares. The line is initially populated with one of each of the numbers from 1 to N. Alice and Bob take turns removing a single number from the line, subject to the restriction that a number may only be removed if it is not bordered by a higher number on either side. When the number is removed, the square that contained it is now empty. The winner is the player who removes the 1 from the line. Given an initial configuration, who will win, assuming Alice goes first and both of them play optimally?

Input

Input begins with a line with a single integer T, $1 \leq T \leq 100$, denoting the number of test cases. Each test case begins with a line with a single integer N, $1 \leq N \leq 100$, denoting the size of the line. Next is a line with the numbers from 1 to N, space separated, giving the numbers in line order from left to right.

Output

For each test case, print the name of the winning player on a single line.

Sample Input 1	Sample Output 1
4 4 2 1 3 4 4 1 3 2 4 3 1 3 2 6 2 5 1 6 4 3	Bob Alice Bob Alice

Problem AB
Pushups
Time Limit: 2

A friend of yours is on the cheer squad for their football team. Each time the team scores, the cheer squad does pushups—one for each point the team has scored so far. If the teams scores a touchdown (7 points), the squad does 7 pushups. If the team then scores a field goal (3 points), the cheer squad does 10 pushups. If the team then scores a safety (2 points), the squad will do 12 pushups. At the end of that game, the squad will end up having done 7+10+12=29 pushups!

You meet your friend after a game, and they say "Boy, am I tired! I did a total of n pushups at the game today!" and promptly collapse from exhaustion. Given n, the number of pushups, can you figure out how the team scored? More than one score may be possible. For example, for 29 pushups, the team could have scored 3, then 2, then 2, then 7, for a total of 14 points. If so, find the highest possible score.

Input

The input will start with a single number on the first line giving the number of test cases, between 1 and 20, inclusive. Each test case will begin with two integers N and M $1 \leq N \leq 5{,}000, 1 \leq M \leq 10$ where N is the number of pushups the cheer squad did, and M is the number of ways a team can score points in that sport. On the next line will be M unique integers S_i, $1 \leq S_i \leq 20$, with a single space between them, indicating the number of points the team gets for each kind of score. The scores are independent; a team can accrue scores in any order.

Output

For each test case, output a single integer indicating the team's final score. If more than one final score can lead to the given number of pushups, output the largest one. If no final score can lead to the given number of pushups, then your friend must have miscounted. In this case, output '`-1`'. Output no extra spaces.

Sample Input 1	Sample Output 1
4 29 3 7 3 2 15 1 1 16 1 1 6 2 3 1	14 5 -1 3

Problem AC
Ritual Circle
Time Limit: 60

Before the departure of the Fellowship from Rivendell, Bilbo gave Frodo his Elvish-made sword that he called Sting. This sword was special: the blade would glow blue whenever Orcs were close.

Input

The input will contain multiple test cases. Each test case will consist of two sets of points in the plane representing the positions of Frodo's companions and the enemy Orcs, respectively. All of these points will be represented by integer coordinates with component values between 0 and 100 inclusive. Every point in each case will be unique. The total number of points from both sets (Frodo's companions and the Orcs) in any single problem instance will be at most 300, and there will be at most 10 test cases with more than 200 points.

Output

Frodo needs to determine the radius of the smallest circle that contains all of the points of his companions' positions, and excludes all of the points of the Orcs' positions. Whenever such a circle does not exist, then the sword Sting glows blue and Frodo is danger, so print "`The Orcs are close`". If such a circle does exist, print the radius of the smallest such circle given as a decimal value that is within a relative error of 1e-7.

In the first example, no circle is possible that includes both companions but excludes both Orcs; any such circle would need to have a radius of at least $\sqrt{1/2}$, but any circle that large would need to include at least one of the Orcs.

In the third example, a circle may be placed with its center an infinitesimally small distance away from $(1/2, 1/2)$ in a direction toward the point $(0, 1)$, with a radius that is infinitesimally larger than $\sqrt{1/2}$.

The fourth example is a degenerate case with only one companion, in which case a circle of zero radius works.

Sample Input 1	Sample Output 1

`Companions: (0,0) (1,1)`
`Orcs: (1,0) (0,1)`
`Companions: (0,0) (0,1) (1,1) (1,0)`
`Orcs: none`
`Companions: (0,0) (0,1) (1,1)`
`Orcs: (1,0)`
`Companions: (0,0)`
`Orcs: none`
	`The Orcs are close`
	`0.707106781186548`
	`0.707106781186548`
	`0`

Problem AD
Saruman Levels Up
Time Limit: 5

Saruman's army of orcs and other dark minions continuously mine and harvest lumber out of the land surrounding his mighty tower for N continuous days. On day number i, Saruman either chooses to spend resources on mining coal and harvesting more lumber, or on raising the level (i.e., height) of his tower. He levels up his tower by one unit only on days where the binary representation of i contains a total number of 1's that is an exact multiple of 3. Assume that the initial level of his tower on day 0 is zero.

For example, Saruman will level up his tower on day 7 (binary 111), next on day 11 (binary 1011) and then day 13, day 14, day 19, and so on.

Saruman would like to forecast the level of his tower after N days. Can you write a program to help?

Input

The input file will contain multiple input test cases, each on a single line. Each test case consists of a positive integer $N < 10^{16}$, as described above. The input ends on end of file.

Output

For each test case, output one line: "`Day N: Level = L`", where `N` is the input N, and `L` is the number of levels after N days.

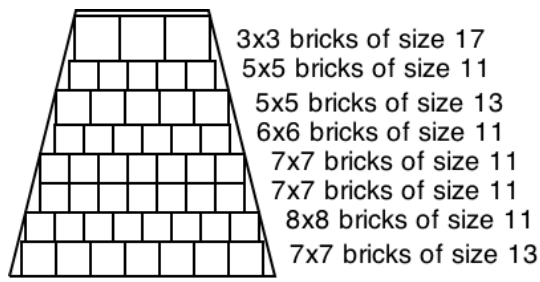
Sample Input 1	Sample Output 1
2 19 64	Day 2: Level = 0 Day 19: Level = 5 Day 64: Level = 21

Problem AE
Temple Build
Time Limit: 2

The Dwarves of Middle Earth are renowned for their delving and smithy ability, but they are also master builders. During the time of the dragons, the dwarves found that above ground the buildings that were most resistant to attack were truncated square pyramids (a square pyramid that does not go all the way up to a point, but instead has a flat square on top).

The dwarves knew what the ideal building shape should be based on the height they wanted and the size of the square base at the top and bottom. They typically had three different sizes of cubic bricks with which to work. Their goal was to maximize the volume of such a building based on the following rules:

The building is constructed of layers; each layer is a single square of bricks of a single size. No part of any brick may extend out from the ideal shape, either to the sides or at the top. The resulting structure will have jagged sides and may be shorter than the ideal shape, but it must fit completely within the ideal design. The picture at the right is a vertical cross section of one such tower.



Height:100; bottom: 100; top: 50

There is no limit on how many bricks of each type can be used.

Input

Each line of input will contain six entries, each separated by a single space. The entries represent the ideal temple height, the size of the square base at the bottom, the size of the square base at the top (all three as non-negative integers less than or equal to one million), then three sizes of cubic bricks (all three as non-negative integers less than or equal to ten thousand). Input is terminated upon reaching end of file.

Output

For each line of input, output the maximum possible volume based on the given rules, one output per line.

Sample Input 1	Sample Output 1
`500000 800000 300000 6931 11315 5000`	`1602937500000000000`

Problem AF
TileCut
Time Limit: 5

When Frodo, Sam, Merry, and Pippin are at the Green Dragon Inn drinking ale, they like to play a little game with parchment and pen to decide who buys the next round. The game works as follows:

Given an $m \times n$ rectangular tile with each square marked with one of the incantations `W`, `I`, and `N`, find the maximal number of triominoes that can be cut from this tile such that the triomino has `W` and `N` on the ends and `I` in the middle (that is, it spells `WIN` in some order). Of course the only possible triominoes are the one with three squares in a straight line and the two ell-shaped ones. The Hobbit that is able to find the maximum number wins and chooses who buys the next round. Your job is to find the maximal number.

Side note: Sam and Pippin tend to buy the most rounds of ale when they play this game, so they are lobbying to change the game to Rock, Parchment, Sword (RPS)!

Input

Each input file will contain multiple test cases. Each test case consists of an $m \times n$ rectangular grid (where $1 \leq m, n \leq 30$) containing only the letters `W`, `I`, and `N`. Test cases will be separated by a blank line. Input will be terminated by end-of-file.

Output

For each input test case, print a line containing a single integer indicating the maximum total number of tiles that can be formed.

Sample Input 1	Sample Output 1
WIIW NNNN IINN WWWI NINWN INIWI WWWIW NNNNN IWINN	5 5