UTSCode

< 2018 >

September 29, 2018

Problems

A Beautiful Primes

B Chocolates

C Enigma

D Limbo: Part 1

E Limbo: Part 2

F Mayhem

G Pipe Rotation

H Polar Express

I Thanos

J Wiseguy

Rules

1. There are **ten** problems to be completed in **5 hours**.
2. All questions require you to read the test data from standard input and write results to standard output. You cannot use files for input or output. Additional input and output specifications can be found in the General Information Sheet.
3. When displaying results, follow the format in the Sample Output for each problem. Unless otherwise stated, all whitespace in the Sample Output consists of exactly one blank character.
4. The allowed programming languages are C, C++, Java, Python2, and Python3. Python 2 is recommended, the PyPy compiler is available and make solutions significantly faster.
5. All programs will be re-compiled prior to testing with the judges' data.
6. Non-standard libraries cannot be used in your solutions. The standard Template Library (STL) and C++ string libraries are allowed. The standard Java API is available, expect for those packages that are deemed dangerous by contestant officials (e.g., that might generate a security violation).
7. Programs will be run against multiple input files, each file containing one or several testcases.
8. Programming style is not considered in this contest. You are free to code in whatever style you prefer. Documentation is not required.
9. All communication with the judges will be handled by the Kattis environment.
10. Judges decisions are to be considered final. No cheating will be tolerated.

Beautiful Primes
Problem ID: beautifulprimes

John Nash is a talented mathematician at Princeton. Due to his prolific contributions in academia, he was even recruited to the Pentagon at a young age to crack enemy encrypted telecommunication. In his first task, John was able to decipher the code mentally, much to the astonishment of other decrypters. In his everyday life, he is constantly looking for patterns in magazines to newspapers to keep his senses sharp.

Recently, he has been cracking away at a cryptography problem that involves prime numbers. Given a positive integer N, the puzzle needs him to produce a "beautiful" list of N primes. A list of N primes numbers is considered beautiful if each prime is at most $1\,000\,000$, and their product has exactly N digits.

For example, if $N = 1$, then the list $[5]$ is beautiful because its length is 1, and its product 5 has 1 digit.

For $N = 2$, the list $[3, 7]$ is beautiful because its product $3 \times 7 = 21$ has the required digit count of 2.

For $N = 3$, the list $[5, 5, 7]$ is beautiful because its product $5 \times 5 \times 7 = 175$ has the required digit count of 3.

John wants to practice his mental math and impress his colleagues. He needs you to write a program that helps him practice this interesting task. John Nash has a beautiful mind, so won't you help him find some beautiful primes?

Input

The first line of input consists of a single integer T ($1 \leq T \leq 50$), the number of test cases.
T lines follow, each of which is a test case consisting of a single integer N ($1 \leq N \leq 1\,000$).

Output

For each test case, print, on a separate line, a list of N space-delimited beautiful primes that are each no greater than $1\,000\,000$. The primes do not have to be distinct, and may be printed in any order. If there are multiple answers, you may print any of them.

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

Chocolates
Problem ID: chocolates

“My mom always said life was like a box of chocolates. You never know what you’re gonna get."

Forrest Gump is a young boy who goes to Greenbow County Central School. As a child, he enjoys running, dancing by swinging his hips, and eating chocolates. Most of all, he enjoys spending time with his best friend Jenny. However, Forrest isn’t the brightest boy by conventional means. While he fully embraces the life wisdom that his mama often imparts (such as through analogies with boxes of chocolates), he still has trouble keeping up with his classes.

Forrest’s math class is currently learning about shapes, specifically polygons. Forrest is falling behind because he doesn’t really understand what a polygon is. Jenny knows that if Forrest doesn’t keep up, his mama would have to take drastic measures to prevent the crooked principal, Mr. Hancock, from transferring Forrest to a special school. As such, Jenny has decided take Forrest’s schooling into her own hands.

Jenny has decided to use something that Forrest understands to explain polygons to him. She picks up a box of square chocolates and empties the pieces onto a napkin, leaving an empty box with R rows and C columns of empty cells. She places a single piece of chocolate in the box and says “*With one chocolate here, I’ve made a square. A square is a polygon, Forrest.*"

Jenny added two more chocolates around the first one and said, “*We still have here a polygon, because we can trace a border around the chocolates without our fingers leaving the surface of the box.*"

Jenny adds some more chocolates, filling up the entire box except a hole in the middle. “*Now Forrest, no matter how we trace the outside box, there will always be a hole we can never draw unless our finger leaves the surface. So this here ain’t a polygon.*"

Jenny removes a chocolate from the corner and says, "*Now we're back to a polygon again! As long as we can trace the border of our chocolates without crossing over where we already traced, we have ourselves here a polygon. As we trace, we can even have two corners barely touch, so long as we don't overlap any border line we've already gone over.*".

"*That's amazing Jenny. Even with just a small box like that, it seems like you can make so many of 'em*", said Forrest.

"*That's right Forrest!*", said Jenny. "*There's so many ways to make a polygon using this box of chocolates, if I just made one at random and had you guess, you truly are never gonna know what you're gonna get!*"

"*Well, Jenny. Just how many ways do you think are there?*" asked Forrest.

"*Hmm, I'm not quite sure about that Forrest.*" Jenny thought for a moment. "*You really have me stumped.*"

Jenny wants to impress Forrest with the answer. Given the dimensions of the chocolate box, can you help her count the number of ways? For example, a 2 by 2 chocolate box has 13 ways of forming a polygon:

```
X.   .X   ..   ..   XX   X.   ..   .X   XX   .X   XX   X.   XX
..   ..   X.   .X   ..   X.   XX   .X   X.   XX   .X   XX   XX
```

Input

The first and only line of input consists of two space-separated integers R and C ($1 \leq R, C \leq 4$), specifying the dimensions of the box of chocolates.

Output

Print, on a single line, the number of different ways that chocolates can form a single polygon in the box. Note that if the same polygon can be placed at multiple different places in the box, then all of those ways are counted separately towards the answer.

Sample Input 1	Sample Output 1
1 2	3

Sample Input 2	Sample Output 2
2 2	13

Enigma
Problem ID: enigma

The Imitation Game is a 2014 film based in 1939 during World War II. It follows the newly created British intelligence agency MI6 as they employ Cambridge mathematics alumnus Alan Turing to crack the German Enigma machine. At the time, cryptographers everywhere believed it to be uncrackable. By the end of the film, Turing and his team were able to successfully crack the Enigma, ultimately winning the war for the allies.

If you have seen the film, you would know that, brilliant as he is, Alan Turing didn't crack the code on his own. He employed help from his group of carefully selected colleagues. But how exactly did he select them? He placed crosswords puzzles all over Great Britain in the hopes that other brilliant minds would be able to solve them to qualify as candidates for his challenging project.

Those who solved his crosswords were given a special test to further evaluate their candidacy. Turing's crosswords were extremely befuddling, but you are smart enough to figure out the words from the clues alone. However, you're not really a spatial thinker, so you need some help positioning them onto the grid. Thanks to the help of modern day computing, perhaps you too can consider yourself worthy to become part of his famous team.

Given an empty crossword puzzle and a scrambled list of all of the solutions to the clues, your task is to position them appropriately on the grid. Like all typical crosswords, words are only valid when read left-right or up-down. Words must start/end either on the edge of the grid or beside a void space.

Input

The first line of input consists of two space-separated integers, R and C ($1 \leq R, C \leq 21$), specifying the number of rows and columns in the crossword grid.

R line follow, each of which consists of C characters, specifying the grid for the unsolved crossword. Within this grid, a "`#`" character represents a void space (i.e. a space where no letters may be placed) and a "`.`" character represents an empty space, where a letter should be placed.

The next line consists of a single integer N ($1 \leq N \leq 200$).

N lines follow, each of which consists of a single string of at least length 2, consisting of only uppercase letters from "`A`" to "`Z`". These lines, given in no particular order, specify every word of the crossword solution that must be placed either horizontally or vertically on the grid.

Output

Print an R by C grid consisting of the solved crossword.
You may assume that there will be exactly one solution in every test case.

Sample Input 1	Sample Output 1
1 15 ##.........#### 1 CROSSWORD	##CROSSWORD####

Sample Input 2	**Sample Output 2**
3 6 #..... ....## ###... 6 AT ME DOG GOD VETO MAGIC	#MAGIC VETO## ###DOG

Limbo: Part 1
Problem ID: limbo1

Dominick Cobb and Arthur are “extractors" who perform corporate espionage. Using experimental military technology that gives them access to shared dream worlds, they infiltrate their targets' subconscious to extract valuable information. Cobb and his entourage are contacted by a mysterious Japanese syndicate and given a seemingly impossible task: instead of extracting information, do the opposite – plant a new idea in the target's mind.

Cobb notes that when using dream-sharing technology, time slows down by different factors depending on whose dream is being entered. Their target is to convince Cepshun, the heir of an energy conglomerate, to dissolve his father's company. Before Cobb starts, he notes that it would be important to figure out the passage of time to ensure that the operation can be completed smoothly.

In Cepshun's dreams, there are multiple different layers of consciousness, each of which controls a different area of Cepshun's dreams and memories. Cobb needs to move between several dreams to reach his destination dream – some of them belonging to Cepshun, and some of them belonging to his entourage. The factor by which time slows down on each level is best visualized by all of the positive integers arranged in an infinite triangle, where the first row has the single integer 1, the second row has the next two integers, the third row has the next three, and so forth. The first 4 levels are depicted below:

```
     1
    2 3
   4 5 6
  7 8 9 10
   etc...
```

The initial time factor for reality is 1. To enter the next dream level, Cobb can either descend left (into a dream by Cepshun) or right (into a dream by his entourage). For example, if he descends right, left, right, then he'll end up at dreams with time factors of 3, then 5, and finally 9. Note the interesting property that for any pair of whole numbers L and R, Cobb will always end up on the same time factor if he moves left L times total and right R times total, regardless of the order in which he makes the moves.

Cobb wants to know the slowness of time inside his target dream, which he can reach by descending left L times and right R times.

Input

The first line of input consists of a single integer T ($1 \leq T \leq 100$), the number of test cases.
T lines follow, each of which is a test case consisting of two space-separated integers, L and R ($0 \leq L, R \leq 10^9$), specifying the number of times Cobb needs to descend left and right, respectively.

Output

For each test case, print, on a separate line, the factor by which time slows down in the target's dream at (L, R).
Note: the answer can be large, and may not necessarily fit in a 32-bit integer.

Sample Input 1	Sample Output 1
3 1 2 2 3 5 8	9 19 100

Limbo: Part 2
Problem ID: limbo2

Dominick Cobb and Arthur are “extractors" who perform corporate espionage. Using experimental military technology that gives them access to shared dream worlds, they infiltrate their targets’ subconscious to extract valuable information. Cobb and his entourage are contacted by a mysterious Japanese syndicate and given a seemingly impossible task: instead of extracting information, do the opposite – plant a new idea in the target’s mind.

Cobb just figured out how to calculate the slowness of time in each dream level, but he is still missing a vital piece of the plan – the architecture. For this challenging mission he has hired the brilliant young architect Ariadne to design a dream space, complete with all the aesthetic and tactile details. Cobb’s target will be brought into that dream space, where they will fill it with details from their own subconscious and memories.

As extractors go into deeper dream levels, the perception of time slows down. This means subjects will be able to explore exponentially more space. Since dreams can go infinitely deep (onwards to Limbo), the architecture of a dream should be infinite in area. The master layout is denoted by a two-dimensional grid that starts at row 0, column 0, and extends infinitely downwards and to the right. Ariadne cannot simply draw a map of this space (since it is infinite), so she must come up with a program that "generates" the dream space to extend as large as the team needs to go deeper. She has an infinite number of building blocks, numbered $0, 1, 2, 3, 4, \ldots$, which must be used in that order.

In dream level 0 (reality), the dream space consists of the single block 0. With each deeper dream level, the area of the space doubles. To prevent the dream space from implosion due to growing too narrowly in a single direction, Ariadne has designed the space to double in *alternating directions*, with new blocks filling in the space sequentially based on the direction of doubling. For example, the first few levels of the dream map are depicted as follows:

```
     0          0 1         0 1       0 1 4 6
                            2 3       2 3 5 7

 (Reality)   (Level 1)   (Level 2)   (Level 3)


   0  1  4  6         0  1  4  6 16 20 24 28
   2  3  5  7         2  3  5  7 17 21 25 29
   8  9 10 11         8  9 10 11 18 22 26 30
  12 13 14 15        12 13 14 15 19 23 27 31

   (Level 4)              (Level 5)
```

Ariadne’s program must be able to generate any part of the map at will. Given only the row and column coordinates for a particular location in the dream space, can you determine the number of the building block that will be used?

Input

The first line of input consists of a single integer T ($1 \leq T \leq 100$), the number of test cases.
T lines follow, each of which is a test case consisting of two space-separated integers, R and C ($0 \leq R, C \leq 10^9$), specifying a particular row and column in the dream-space.

Output

For each test case, print, on a separate line, the number of the building block at coordinates (R, C).
Note: the answer can be large, and may not necessarily fit in a 32-bit integer.

Sample Input 1	Sample Output 1
3 0 0 1 0 2 4	0 2 18

Mayhem
Problem ID: mayhem

You are a seasoned Fight Club member who recently stumbled upon the classified plans of Project Mayhem, a secret organization that the other members have created to bring down consumerism and corporate America. To completely destabilize modern civilization, Tyler Durden and his Project Mayhem trainees are making plans to take down the buildings of major banks that hold everybody's credit information. If they succeed, then everybody is supposed to restart life with a blank slate, free of all debts ranging from mortgages to student loans.

Tyler plans to do so by setting off pipe bombs under major financial buildings. His crew has mapped out the city's financial district on a rectangular grid with R rows and C columns. Each grid cell either has a building armed with bombs to detonate (denoted by "`x`") or does not (denoted by "`.`"). As a friend of Tyler's, you know all too well that his plan is off the chains. Violence is never the solution. You want to do your best to thwart the plan by visiting the buildings one by one and disarming as many bombs as possible.

You can disarm the buildings in any order, with only one catch. After you disarm a building, Tyler's crew will find out, chase after you, and occupy that building indefinitely. To avoid getting caught, you must immediately find another (armed) building in the same street or avenue to take cover before disarming other buildings. After that however, you can go to any other armed building on the map to disarm it (provided the same condition is met). In other words, at any time you disarm a building, there must be at least one other armed building in either the same row or the same column.

Take the following 3 by 3 map for example:

```
x..
.x.
x.x
```

You have a few ways to disarm up to 2 buildings:

- You can first disarm the top-left one, followed by either of the bottom two buildings, for a total of 2 buildings.
- You can first disarm the bottom-right one, followed by either of the remaining two leftmost buildings, for a total of 2 buildings.
- However, note that if you choose to first disarm the bottom-left building, then none of the remaining buildings can be disarmed (since you won't be able to take cover afterwards). So in this case, you would only be able to disarm a single building.

In any of these cases, the center building can never be disarmed since there is no immediate neighboring building in its row or column to use for cover from Tyler's henchmen.

Given a map of the city, you would like to know the maximum number of buildings that can be disarmed.

Input

The first line of input consists of two space-separated integers R and C ($1 \leq R, C \leq 2\,000$).
R lines follow, each of which consists of C characters (either "`x`" or "`.`"), specifying the grid of the financial district.

Output

Print, on a single line, the maximum number of buildings that can be disarmed.

Sample Input 1

```
3 3
x..
.x.
x.x
```

Sample Output 1

```
2
```

Sample Input 2

```
3 4
.xx.
x...
x..x
```

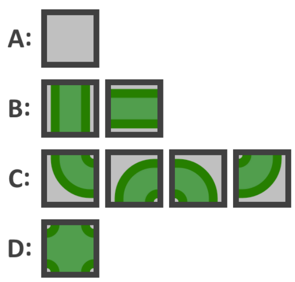
Sample Output 2

```
3
```

Pipe Rotation
Problem ID: piperotation

The four ninja turtles: Leonardo, Donatello, Michelangelo, and Raphael are seeking a new home in Manhattan, New York City. The turtles don't like sudden dead-ends in their home. Fortunately, the government recently installed a new sewage system where pipes can be rotated! The turtles needs your help finding a suitable home, so they're willing to provide you a grid of the current layout of the sewage system.

The grid consists of N rows and M columns. The cell $G_{i,j}$ will consist of one of four pipes, encoded as a letter between "`A`" and "`D`". These pipes can be rotated by any multiple of 90 degrees:



- (`A`) Nothing
- (`B`) Straight pipe (pipes leaving through two opposite edges)
- (`C`) Elbow-shaped pipe (pipes leaving through two adjacent edges)
- (`D`) Four-way pipe (pipes leaving through all four edges)

Determine whether it's possible to rotate the cells such that the pipes all line up with one another. In particular, for each edge shared by a pair of adjacent cells, there must either be a pipe on both sides of that edge, or on neither side. And for each each of the $2 \cdot (N + M)$ outer edges of the grid, there must be no pipe leaving through that edge. Below are examples:

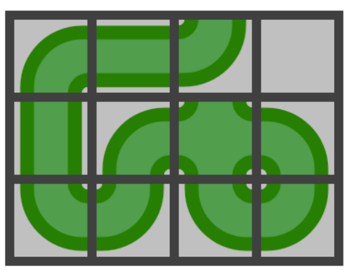


Fig 1. Invalid example,
two sudden dead-ends.

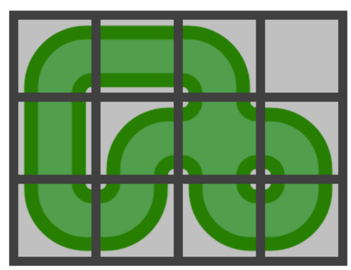


Fig 2. Valid example,
no sudden dead-ends.

Input

The first line of input consists of two space-separated integers, N and M ($1 \le N, M \le 100$).
N lines follow, the ith of which consists of M characters $G_{i,1}, G_{i,2}, G_{i,j}, \ldots, G_{i,M}$ and $G_{i,j} \in \{A, B, C, D\}$ (for $i = 1..N$).

Output

Print, on a single line, a single string, either "`Possible`" if it's possible to produce a valid configuration of pipes, or "`Impossible`" otherwise.

Sample Input 1	Sample Output 1
2 2 CC CC	Possible

Sample Input 2

```
2 2
CC
CB
```

Sample Output 2

```
Impossible
```

Sample Input 3

```
3 3
CCC
CCC
CCC
```

Sample Output 3

```
Impossible
```

Sample Input 4

```
3 4
CBCA
BCDC
CCCC
```

Sample Output 4

```
Possible
```

Sample Input 5

```
5 2
CC
CC
AA
CC
CC
```

Sample Output 5

```
Possible
```

The Polar Express
Problem ID: polarexpress

It's almost December, and the Polar Express is planning for its annual trip towards the North Pole. The Polar Express is a train with N carts numbered 1 to N, and you are the conductor! However, due to an exceptionally heavy snowstorm, your train recently ran into an accident and veered off-track. Now, the ordering of all of the carts have been scrambled! You cannot leave to pick up the children for Christmas unless you can reorganize the train in the original sorted order.

At the start of every hour, you can move the first K carts from the front of the train to a holding station, which is a stack. At the end of every hour, the magical elves of the holding station will move the carts from the holding station to the back of the train, in reverse order.

For example, suppose the train is currently ordered $1, 2, 5, 4, 3$ from front to back. You can move the first three trains to the holding station. At the end of the hour, your carts will be returned to you, and the train will be ordered as $4, 3, 5, 2, 1$.

Your job is to come up with a sequence of instructions to sort the train, such that cart 1 is at the front and cart N is at the back. However, Christmas is in $1\,000$ hours and you cannot take longer than that to complete your task.

Input

The first line of input consists of a single integer N ($1 \leq N \leq 200$), specifying the number of carts that the train contains.
N lines follow, each of which consists of a single integer, together forming a permutation of the numbers from 1 to N. This permutation specifies the initial order of the train carts.

Output

On the first line of output, print a single integer M, the number days it will take to sort the train.
Print M lines next, the ith of which should consist of a single integer denoting the number of carts to move to the holding station on the ith hour. Each integer printed must be between 1 and N, inclusive.
Your answer will be judged correct if $0 \leq M \leq 1\,000$, and your given sequence correctly sorts the train carts.

Sample Input 1	Sample Output 1
5 3 2 1 4 5	2 4 2

Thanos
Problem ID: thanos

Dr. Thanos, data scientist and self-described nihilist, recently published a paper titled *The snapping point of the universe: why rates of population growth point towards imminent destruction*. In his paper, Thanos explains that in many planets, the increasing population count is leading to a diminished quality of life. He is convinced that his findings will drive sweeping reforms in intergalactic law, leading to a better life for all organisms.

Thanos turns to you, his confidant, to do some investigation. He would like some concrete evidence for his findings to present to the Association of VENGE's Research Society. The society, one of the galatic leaders in egalitarianism and social justice, is holding a special panel to discuss Thano's findings. As this involves the chance of actual legislation being passed, Thanos is convinced that the panelists are going to be a tough sell. He asks you to investigate several datasets and see if they could be potentially helpful in supporting his argument.

Thanos hands you the data for several planets. On each planet's file, you read that: the planet currently has a population of P, its population grows by a factor of R times per year, and its annual food production F in tons. Assume that each individual consumes 1 ton of food per year, and that the population for each planet each year is always counted as a whole number, rounded down. Given this information, your task is to find out the number of years a planet has remaining before its population is no longer sustainable by its food production.

Input

The first line of input consists of a single integer T ($1 \leq T \leq 2\,000$), the number of planets that need to be analyzed. T lines follow, the ith of which consists of three space-separated integers P ($1 \leq P \leq 10^9$), R ($1 < R \leq 10^9$), and F ($1 \leq F \leq 10^9$), the metrics of planet i as described above.

Output

Print T lines, the ith of which should consist of a single integer denoting the number of years the ith planet has before it is no longer sustainable.

Sample Input 1	Sample Output 1
3 1 3 9 2 2 16 5 2 11	3 4 2

Wiseguy
Problem ID: wiseguy

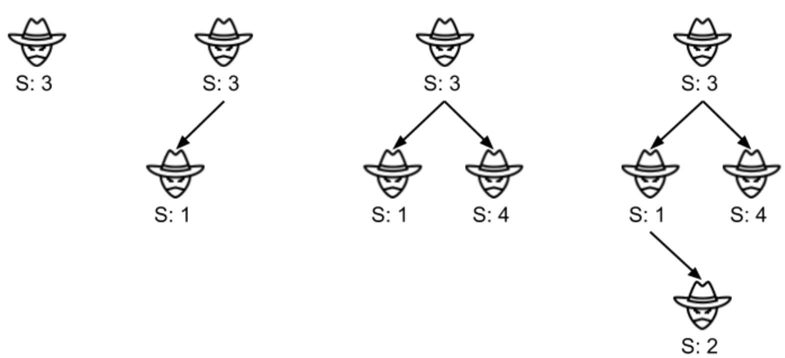
Paulie Cicero is a made man who runs the underworld of Brooklyn along with his associates Jimmy "the Gent" Conway, Tommy DeVito, and the young but ambitious Henry Hill. The crew just got wind of a one-in-a-lifetime opportunity to raid $6 million from the Lufthansa vault at John F. Kennedy International Airport. If they plan carefully and succeed, this will go down as the greatest heist of all time.

Jimmy plans to hire N new recruits for the operation, whom he plans to organize into a hierarchy of leadership. Recruits arrive one-by-one on a rolling basis. The first recruit has no leader, but every subsequent recruit that comes in must immediately be assigned to exactly one boss (who must be some previous recruit already in the hierarchy). Henry points out that for things to go smoothly, each recruit should in the end only be assigned 0, 1, or 2 subordinates – either a left-hand man, or a right-hand man, or neither, or both. Setting a limit at 2 will reduce each individual's responsibility of having to manage lots of people.

Each recruit also has a distinct "strength" level (say, as a distinct integer from 1 to N), which Jimmy will evaluate and consider as he is assigning leadership. We can assume that the strength levels across all of the recruits are uniformly random, and that recruits arrive in no particular order of strengths. That is, the final sequence of strengths among the N recruits ordered from oldest to newest is drawn uniformly randomly from the set of all $N!$ possible permutations.

Paulie knows from experience that having a strong recruit leading other weaker recruits is a bad idea (the boss might abuse the subordinates), as is having a weak recruit leading only stronger recruits (being more qualified than your boss can lead to insubordination). Henry came up with a simple rule to mitigate this problem and bring some balance: For each recruit i, the left-hand man (if he exists) should always be weaker than recruit i himself. Conversely, the right-hand man (if he exists) should always be stronger than recruit i himself.

Each new recruit that comes in is first handed to the first recruit (except the first recruit himself). Based on Henry's rule, the new recruit is passed down to become either the left- or right-hand man of the current recruit that has custody of him. If there is already a left- or right-hand man, then the recruit is passed down further. This repeats until there are no conflicts and the new recruit is settled in as a subordinate of an existing recruit who previously only had either 0 or 1 subordinates. The diagram below illustrates the boss assignment process for $N = 4$ recruits, who arrive in the following order of strengths: 3, 1, 4, 2.



While Henry's strategy of limiting the number of subordinates to 2 per person is good for distributing responsibility, the hierarchy can also become quite "vertical". Having excessively long chains of command can lead to miscommunication and broken telephones, which the crew cannot afford in a delicate operation like this. The "height" of a hierarchy is defined as the maximum number of times that a message needs to be passed from the first recruit to be able to reach any other recruit in the hierarchy. For example, the height of a hierarchy with a single leaderless recruit is 0, and the height of the last hierarchy in the above diagram is 2.

Given the uniform randomness of new recruit strengths as well as the rules above for assigning leaders, Henry needs your help in finding out the expected height of the final hierarchy. This can be thought of as the "average" height of hierarchies across all possible arrival orders. For example, in the case of having to organize $N = 2$ recruits, both possible hierarchies have a height of 1. In the case of $N = 3$, two of the possible hierarchies have a height of 1, and the remaining four possible hierarchies have a height of 2, resulting in an expected height of $(1+1+2+2+2+2)/6 = 5/3 \approx 1.66667$.

Please help Henry predict the height for a hierarchy of N recruits, so he can decide whether the plan will be feasible. A true wiseguy never gets caught, but a bad judgment or miscommunication in the ranks can easily bring shame to your crew. You'd better not mess up, or you might as well listen to Billy Batt's advice for Tommy – quit the mob life, go home and get your shine box.

Input

The first line of input consists of a single integer T ($1 \leq T \leq 500$), specifying the number of test cases to follow. T lines follow, each of which is a test case consisting of a single integer N ($1 \leq N \leq 500$), specifying the number of recruits that will need to be organized for the heist.

Output

For each test case, print, on a separate line, a single real number denoting the expected height for the hierarchy. Note: your answer must have at most 10^{-5} absolute or relative error to be considered correct.

Sample Input 1	Sample Output 1
2 2 3	1.00000 1.66667