[**Problem 1: Boomer SooNer**](#_es68slnjbiqs) **2**

[**Problem 2: Program Discipline**](#_w7xlgq8a1yf7) **3**

[**Problem 3: Mister Money Miser the Maximizer**](#_95kjmrhms00n) **4**

[**Problem 4: Be Sure to Drink Your Ovaltine!**](#_ykphssadchct) **5**

[**Problem 5: Roll for Initiative**](#_9atddzx5odj6) **6**

[**Problem 6: Old McDonald Had some Glorps**](#_56bu7x2k202l) **7**

[**Problem 7: TOP 10 CHEEKY PROBLEM DISGUISES! (YOU WON’T BELIEVE #7)**](#_77nqiervnk7y) **8**

[**Problem 8: The Frightening Plight of the Nine Knights’ Spite**](#_dg27ynuzt9zz) **9**

[**Problem 9: Once Upon A\***](#_dy3wzwgtutdk) **10**

# Problem 1: Boomer SooNer - kevin

A tradition at OU football games is the “Boomer! Sooner!” chant, where one of the stadium yells “Boomer!” and the other responds “Sooner!”. This is a great way to hype up the crowd and to get in the opposing team’s head. However, it can sometimes be awkward for the crowd when they don’t know when to stop. They just keep yelling “Boomer!” “Sooner!” “Boomer!” “Sooner!” “Boomer!”… until one side finally gets quiet enough that the other side doesn’t hear them. OU President James Gallogly has asked you to solve this problem. He wants to give you a number 0<N<1000 and have you put “Boomer!” followed by “Sooner!” on the big screen N times. Can you do it?

**Input:**

The first line will contain the integer 0<N<1000.

**Output:**

Print the “Boomer!” followed by a newline followed by “Sooner!” followed by another newline N times.

**Example Cases:**

|  |  |
| --- | --- |
| Input | Output |
| 1 | Boomer!  Sooner! |
| 5 | Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner! |

Hint 1: Remember the Newline at the end of the last “Sooner!” Also recall that some print functions automatically include newlines and some don’t.

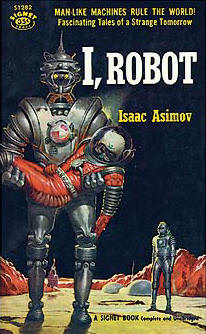
1: Boomer SooNer. Judges Data

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 | Boomer!  Sooner!  Boomer!  Sooner! |
| 15 | Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner!  Boomer!  Sooner! |

# 

# Problem 2: Program Discipline (RYAN)

The programs at the Military Academy of Computer Programs have been behaving badly lately, executing commands on their own and sending questionable emails to Academy staff. As I’m sure you know, the Academy doesn’t tolerate disobedience. We need a program that demonstrates good behavior by doing what it’s told.



The second law of robotics: “A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.”

**Input:**

The only line will be text of the form “Do <command>!”

**Output:**

Print “Did <command>!” followed by a newline.

**Example Cases:**

|  |  |
| --- | --- |
| Input | Output |
| Do 20 pushups! | Did 20 pushups! |
| Do some thinking about your behavior! | Did some thinking about your behavior! |

# Problem 3: Mister Money Miser the Maximizer -David

Mister Money Miser did pretty well in school, so he gets a lot of job offers. When choosing which offer to accept, he doesn’t consider work-life balance or perks, and he sure as heck doesn’t care about the workplace culture. No, Mister Money Miser only looks at one thing: the moolah.

Your job is to help Mister Money Miser out by writing a program that takes five potential salaries and prints the highest one.



“My money’s the best friend I ever had”

**Input:**

The first five lines will each contain a single integer 1000<N<100000.

**Output:**

Print the greatest of the five input integers followed by a newline.

**Example Cases:**

|  |  |
| --- | --- |
| Input | Output |
| 2000  20000  1500  95000  12345 | 95000 |
| 8000  8888  4321  4321  8888 | 8888 |

# Problem 4: Be Sure to Drink Your Ovaltine!

In a famous scene in the classic movie A Christmas Story, Ralphie, after drinking an untold number of gallons of ovaltine in order to rack up enough points to earn a special decoder, tunes in to the weekly “Little Orphan Annie” show, writes down ‘Annie’s Secret Message’, a series of numbers, and uses the decoder to turn those numbers into a phrase. In the movie, the phrase turns out to be an add for even more ovaltine, but we assure you our messages are far more important.

Your task is to read in N numbers and output the decoded phrase. We know you probably don’t have access to near as many gallons of ovaltine as Ralphie, so we provided you with a special “top secret” decoder below. Make sure not to share it with anyone else

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| q | m | l | j | v | t | r | g | p | u | o | f | b | z | a | i | n | w | h | x |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 21 | 22 | 23 | 24 | 25 | 26 |
| k | s | y | e | d | c |

**Input:**

The first line will be an integer 0<N<1000. The next N lines will be integers 1<=N<=26, each corresponding to a letter.

**Output:**

Print the decoded letters. No spaces or newlines necessary. All letters lowercase

**Example Cases:**

|  |  |
| --- | --- |
| Input | Output |
| 2  19  16 | hi |
| 10  16  3  16  21  24  9  16  14  14  15 | ilikepizza |

# Problem 5: Roll for Initiative - kevin

You are the Game Master for a Dungeons & Dragons campaign with some of your friends, and you have an unhealthy obsession with statistics and probability. You tell all your players to roll their dice, but you specify which die each of them should use. You want to be able to anticipate their rolls by calculating the expected average value of the entire party’s dice.

We will assume each player has obtained a complete collection of dice with all possible numbers of sides, which they keep in their Bag of Holding. Each player might be assigned dice with different numbers of sides, or there could be overlap among party members. Dice are numbered starting from 1 and increasing up to their max value, which equals the number of sides. Since you are friends with everyone in the party, they will use the dice you tell them to.

**Input:**

The first line will contain a single integer 0<N<10000 specifying the number of party members who will be rolling. The next N lines will contain an integer 1<S<10000 denoting the number of sides on the die they will roll. (e.g., S = 6 would denote rolling a d6, or a 6-sided die)

**Output:**

A single number accurate to 3 decimal places representing the average of the average values of each individual die. The number should be rounded to 3 decimal places, not simply truncated.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Example Cases:**   |  |  | | --- | --- | | Input | Output | | 5  6  6  6  6  6 | 3.500 | | 3  4  8  20 | 5.833 | |  |

Hint 1: To output to 3 decimal places, look over the section in the API for your chosen language about print formatting.

Hint 2: The average value of each die can be easily found by taking (S+1)/2. Then the output will simply be the average of these numbers.

5: Roll for Initiative. Judges Data

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2 | 1.5 |
| 3  7  7  7 | 4 |
| 6  3  8  9  20  2  8 | 4.667 |

# Problem 6: Old McDonald Had some Glorps -David

Farmer Hjalmar McDonald is doing his yearly Glorp planting, but has forgotten the arrangement from the year before. For those who didn’t pay attention in Glorps 101, Glorps are highly valuable plants that, if planted too close to one another, will eat each other when they have grown. Old Man Hjalmar obviously would like to avoid this, and has asked you to check his proposed new arrangement.



A Glorp. Definitely not whatever copyrighted character you think it is.

The input for this problem consists of a number of lists of coordinates, followed by lists of coordinates for each location Hjalmar proposes planting his Glorps. Your job is to determine, given these proposed locations, whether Hjalmar’s plants will all be safe (or whether they’ll eat each other!). Plants must be at least more than 1 unit apart from each other, and coordinates are given in the form x1 y1 x2 y2 x3 y3..., where xn is the nth x-coordinate and yn is the nth y-coordinate. All numbers given will be non-negative and will be storable as doubles.

|  |  |
| --- | --- |
| Input | Output |
| 3  1 1  2 2  3 3  3  1 1  1.5 1.5  2 2 | Safe  Unsafe |

# Problem 7: The Most Scientific Study to Ever be Studied by Studiers

Somebody accidentally let OU Computer Science Majors into the Genetics Labs, and while working on a project for Machine Learning, they accidentally messed with all of the population parameters of the rabbit breeding study! Thankfully, the geneticists working in the lab weren’t mad at the CS students for ruining the study. In fact, they were very impressed that these students had managed to write code so terrible that it altered the behaviors of their rabbits! After several weeks of debates among the higher-ups in control of the Genetics Labs, it was decided that the CS students would not be punished for the damage they caused if they could predict the total number of rabbits after N generations of breeding these genetically altered rabbits within a week. That was 6 days ago, and the students have come to you to help them figure it out!

The CS students have managed to figure out three key things about these rabbits. First, each adult rabbit can now magically generate 1 baby rabbit per breeding cycle, all on it’s own! Second, the baby rabbits become able to generate new rabbits one generation after they are born. Lastly, due to a massive bug in the students’ code, once a rabbit has generated its second offspring (which happens in the second generation in can reproduce), it immediately dies. The students have noted that the rabbit population seems to grow increasingly quickly, especially since each rabbit is producing 2 more, but they can’t come up with an equation for the number of rabbits alive during each specific generation. That’s why they’ve come to you!

To check their assertions, you start a population (generation 1) with 1 genetically altered baby rabbit. In the next generation (gen 2), that rabbit is an adult now and produces a child, so there is 1 adult rabbit and 1 child bunny (2 rabbits). In the next generation (gen 3), the adult bunny is now a grandBunny, and it produces a child and promptly dies. The child from the last generation is now an adult and also creates a bunny. Now there is 1 adult rabbit and 2 child rabbits. Having remembered what your math teacher told you about series, you assert that the next generation will have 5 rabbits, and you tell the nervous CS majors how to easily predict the number of rabbits alive during the nth generation.

**Input:**

The first line will be an integer 1 <= N < 2^31 corresponding to the number of generation for which you want to know the number of rabbits.

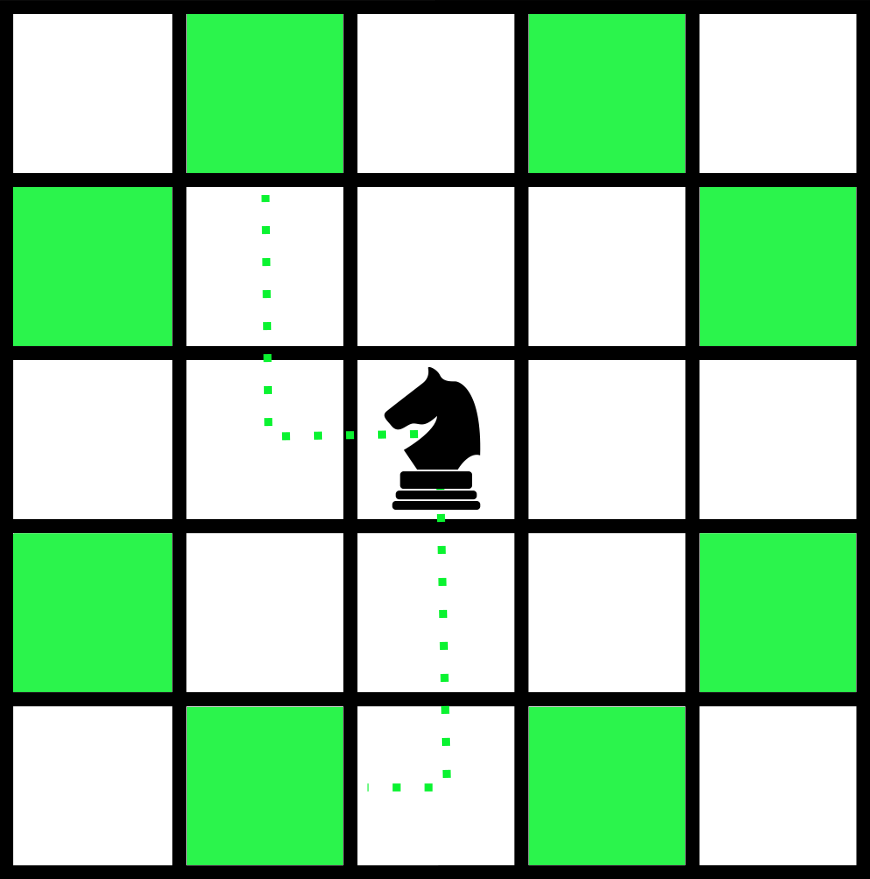
**Output:**

Print the number of rabbits alive during that generation

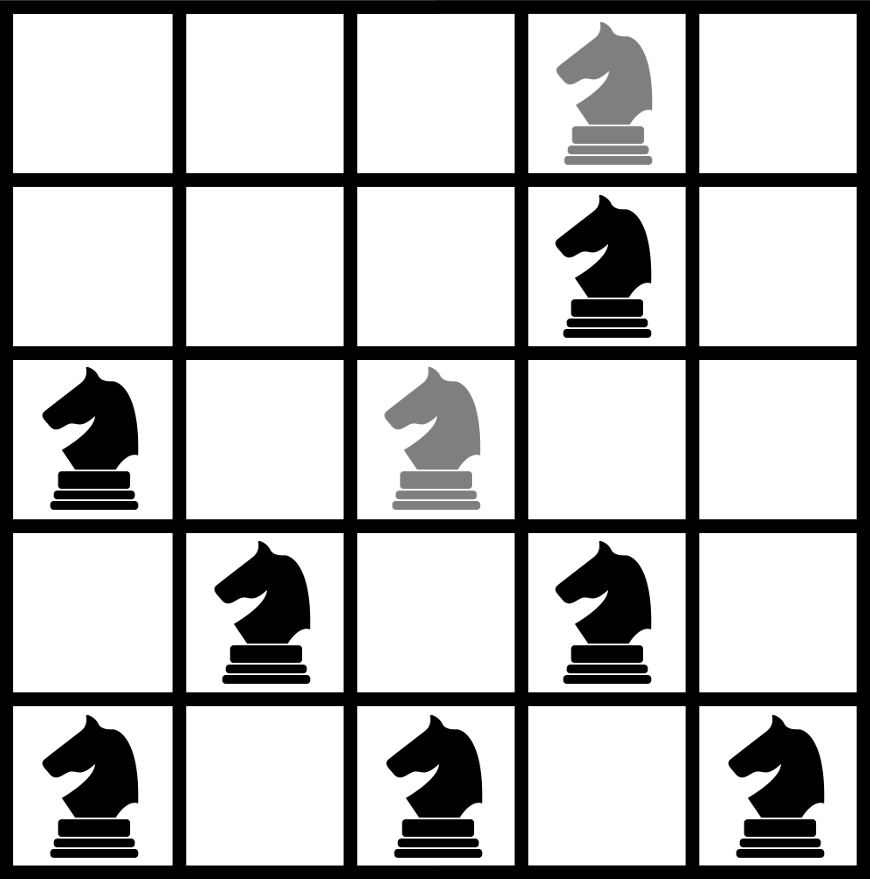
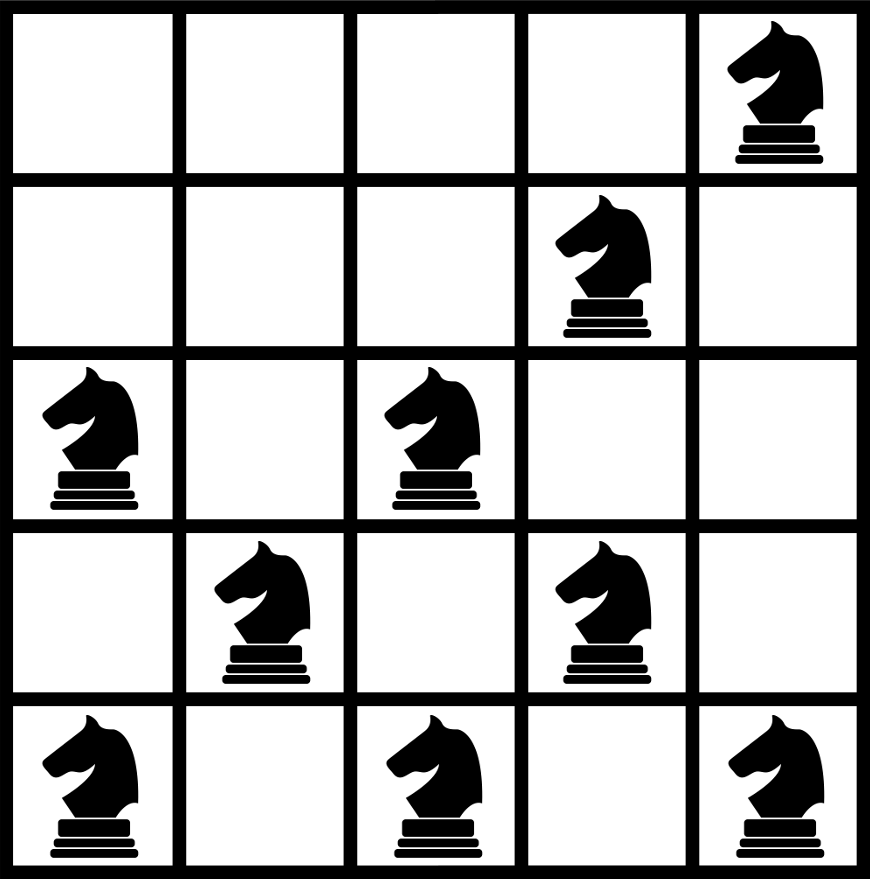
**Example Cases:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |

# Kevin- Problem 8: The Frightening Plight of the Nine Knights’ Spite (copy of nineknights problem on Kattis)

In the game of chess, knights are unique due to their “L-shaped” movement. A knight can move, as shown in Figure 1, by either moving two squares sideways and one square up or down, or moving one square sideways and two squares either up or down.

**Figure 1**: The highlighted squares show all possible moves for a knight.

In the Nine Knights puzzle, exactly nine knights must be positioned on a 5-by-5 board so that no knight can attack another knight with a single move. The configuration shown in Figure 2 is an invalid solution because two of the knights can attack each other, where the configuration shown in Figure 3 is a valid solution.

**Figure 2**: Invalid game configuration **Figure 3**: Valid game configuration

Given the description of a game configuration, your job is to determine whether or not it represents a valid solution to the Nine Knights puzzle.

**Input:**

The input will consist of 5 lines, each having 5 characters. All characters will be either ’k’, indicating the placement of a knight, or ’.’, indicating an empty space on the board.

**Output:**

Display the word valid if the given chess board is a valid solution to the Nine Knights puzzle. Otherwise, display the word invalid.

**Example Cases:**

|  |  |
| --- | --- |
| Input | Output |
| ...k.  ...k.  k.k..  .k.k.  k.k.k | invalid |
| .....  ...k.  k.k.k  .k.k.  k.k.k | valid |
| .....  ...k.  k.k.k  .k.k.  k...k | invalid |

8: The Frightening Plight of the Nine Knights’ Spite. Judges Data

|  |  |
| --- | --- |
| **Input** | **Output** |
| k....  ...k.  k.k..  .k.k.  k.k.k | valid |
| kkkkk  .....  .....  .....  kkkk. | valid |
| kkkkk  kkkk.  .....  .....  ..... | invalid |
| k..k.  .k...  ..k..  ...k.  kkkk. | invalid |

# 

# Problem 9: Proof by Staircases

On your way to math class, you face a staircase which seems to have a different number of steps every day, and one day you decide you are interested in just how many different ways there are to climb the stairs, given the total number of steps. Since you have long legs, you can take the steps two at a time or one at a time, which means there is certainly more than one way to climb the stairs.

**Input:**

A single integer 0<N<10000 denoting the total number of steps on the staircase.

**Output:**

A single integer denoting the number of possible ways to climb the stairs taking either one or two steps at each step.

**Example Cases:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 | 1 |
| 2 | 2 |
| 6 | 13 |