2021 UMW HS Programming Contest

Overview[1]
Welcome to the UMW High School Programming Contest - 2021! This is a contest wherein teams from area high schools compete to solve a set of programming problems as fast as possible.

Problems
There are 14 problems of (approximately) increasing difficulty. The team who solves the most problems is declared the winner. In the event of a tie, the team with the best time wins. The best time includes penalties for wrong submissions. Each problem has a general description, details on the formats for input and output, and sample input and output.

Input is from standard input and output is to standard output. Do not open any files inside of your program and do not prompt for input. Output of your program must match that of the correct output ***exactly***. You can solve the problems in Java, Python, or C++. Up to date versions of these languages are used to judge your entries. You must save your code using the standard extension for your language of choice (.java .py .cpp respectively). In case you need a refresher, sections at the end of your packet provide a brief overview of how to perform input and output in Java, Python, and C++.

Rules
You are allowed to have any printed material such as books or printouts. You are also allowed to access standard library documentation for your language of choice. No other use of the internet is permitted. You are also not allowed to copy and paste code from any source, be it a thumb drive, or another file that you have. Since the contest is virtual, most contestants will be using a computer to participate. Each team is allowed to concurrently use two of the computers to write code. The other teammates cannot use computers to write code.

Logging In
To log in to the contest system, direct your browser to <URL TBD>, and enter your team name and password provided. This brings up the interface for you to submit solutions to the problems, ask questions (which are broadcast to all participants), and see how your team is doing.

Submitting
To submit a solution, first make sure that you select the correct problem letter on the top of the site. Then click the "Choose File" button. Then choose the file that contains your source code (be sure it has the correct extension). After that it will be submitted for judging. It will receive an automatic judging response which is marked pending. Your solution will then be checked manually and a final response will be assigned.

[1] Overview page modified from Dr. Finlayson's 2018 UMW Contest

Introduction

Welcome to the UMW CPSC High School Programming Contest in 2021. We hope that you have fun. If you have questions during the contest, please ask. There is not a specific metric for easy / hard, but the problems are generally ordered from easiest to hardest.

The scene of this year's contest is the quaint town of Maryburg, which is home of Youemdubya College. Maryburg has a port that harbors the famous pirate ship Jaffyboat. For the most part, Maryburg townies, Youemdubya students, and Jaffyboat pirates get along. Youemdubya students contribute to Maryburg's population and Jaffyboat pirates contribute to Maryburg's economy. The problems include various tidbits about life in Maryburg.

- The Census workers from Deecee conduct a strange census in Maryburg.
- There is a treasure at the top of a square spiral staircase.
- Jaffyboat pirates brawl at the local saloon.
- Youemdubya Cyber Security majors created an encryption app for Maryburg's townsfolk.
- Maryburg's Conery Creamery has famous triple scoop ice cream cones.
- Maryburg's elementary students learn math by playing Fizzle Buzzle Puzzle.
- Youemdubya science students create toilet paper from eucalyptus leaves and sodium chloride to alleviate shortages caused by COVID19.
- PiQueue, the supply officer on the pirate ship Jaffyboat, goes on two shopping sprees in Maryburg.
- The Jewelry Cube in downtown Maryburg has diamonds for sale.

Maybe the problems and tidbits will influence you to visit Maryburg and Youemdubya, which is a wonderful college with an outstanding computer science department.

Problem A - **Practice** - How Old is Jaqarue

Jiffy and Jaffy are residents of Maryburg. They have an uncle named Jaqarue. His given name is Jaque, but in college he did extensive field work in Australia studying kangaroos, where he received the nickname Jaqarue. Jaqarue's speciality is Red Giant Kangaroos. Jaffy is 19 years old and a sophomore at Youemdubya, the college in the middle of Maryburg. Uncle Jaqarue is 49 years old. Write a program that given Jaffy's age, the program prints Uncle Jaqarue's age.

Input

Input is a single, non-negative integer that is Jaffy's age.

Output

Output is one line that is the age of Jaqarue.

Sample Input 1

```
10
```

Sample Output 1

```
Uncle Jaqarue is 40 years old.
```

DO NOT SCROLL DOWN UNTIL THE CONTEST STARTS

Problem B - Maryburg's Population

The town of Maryburg is quaint, but the population is growing. The only reason for growth is the beautiful college Youemdubya College located in the middle of town. Many students love Maryburg and they decide to settle in Maryburg after they graduate. Youemdubya College always has 2,000 students, equally divided in freshman, sophomore, junior, and senior. The current year is 2021, and the town of Maryburg's population is currently 10,000. The only population growth in Maryburg is the 10% of Youemdubya's graduating seniors. Two years ago the population of Maryburg was 9,900. You must write a program that provides the population of Maryburg in a given year.

Input

Input is a single, integer that is the year for the population.

Output

Output is a single, non-negative integer that is the population of Maryburg corresponding to the year.

Sample Input 1

```
2024
```

Sample Output 1

```
10150
```

Sample Input 2

```
2031
```

Sample Output 2

```
10500
```

Problem C - Pirate Saloon

Maryburg has a local saloon that the pirates love to visit when their famous pirate ship, Jaffyboat, is in port. Most of the time the pirates are calm when visiting the saloon, but on this night they got involved in a vicious brawl. Fortunately they fought among themselves so no

residents of Maryburg or students at Youemdubya were hurt. The captain hauled all but two of the pirates back to the ship. Two pirates, Jessabell and Dinglebell, did not fight so they remained in the saloon playing their favorite word game - Longest Word. Jessabell and Dinglebell are former English majors at Youemdubya. They know a lot of words. The game consists of each person spelling a word. The longest word wins. You determine the winning word.

Input

Input is two lines. Each line has a word.

Output

Output is one line, which is the longer of the two words or the word TIE if both words are the same length.

Sample Input 1

hello
pirateship

Sample Output 1

pirateship

Sample Input 2

hello
gusty

Sample Output 2

TIE

Problem D - Maryburg and COVID19

Like all communities, the town of Maryburg endured COVID19 hardships. One hardship involved toilet paper. The grocery stores ran out of toilet paper. The Youemdubya College biology and chemistry majors collaborated to create toilet paper for the entire campus. Leaves from the eucalyptus trees on campus were ground into a powder, mixed with sodium chloride, pressed into squares of 4.5 inches, which after drying resulted in a soft experience similar to Sharman. It took several attempts, but eventually the mixture of 3 ounces of eucalyptus leaves

and 4 ounces of salt produced 10 toilet paper squares. You can only make whole toilet paper squares. Ten eucalyptus leaves can be ground into 3 ounces. You must write a program that shows the number of sheets of toilet paper given quantities of eucalyptus leaves and sodium chloride.

Input

Input is one line with two, non-negative integers. The first integer is the number of eucalyptus leaves. The second integer is the number of ounces of sodium chloride.

Output

Output is a single integer, which is the number of sheets of toilet paper created by the input.

Sample Input 1

```
30 12
```

Sample Output 1

```
30
```

Sample Input 2

```
45 100
```

Sample Output 2

```
40
```

Problem E - Fizzle Buzzle Puzzle

You are the elementary math teacher in the small town of Maryburg. You have a group of energetic children who want to learn division. At first you are puzzled about how to help them, but then you remember the Fizzle Buzzle Puzzle game from your youth. Fizzle Buzzle Puzzle can be played by many children at once, and it uses numbers and the words `Fizz` and `Buzz`. Children take turns to counting numbers aloud - one, two, etc. Numbers divisible by three are replaced with the word `Fizz`, and numbers divisible by five are replaced with the word `Buzz`. They would count aloud - `1`, `2`, `Fizz`, `4`, `Buzz`, `Fizz`, `7`, `8` and so on. Create a program that iterates through numbers and prints what the children should count aloud.

Input

Input is a single, non-negative integer that is the number of times to be sequenced using the Fizzle Buzzle Puzzle game.

Output

Output is a line or lines of strings of integers and either "Fizz', "Buzz", or "Fizz Buzz" separated by commas and spaces. The String should end with a period. Each line (except for the last) must contain ten instances of number, Fizz, Buzz.

Sample Input 1

```
15
```

Sample Output 1

```
1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz,
11, Fizz, 13, 14, Fizz Buzz.
```

Sample Input 2

```
36
```

Sample Output 2

```
1, 2, Fizz, 4, Buzz, Fizz, 7, 8, Fizz, Buzz,
11, Fizz, 13, 14, Fizz Buzz, 16, 17, Fizz, 19, Buzz,
Fizz, 22, 23, Fizz, Buzz, 26, Fizz, 28, 29, Fizz Buzz,
 31, 32, Fizz, 34, Buzz, Fizz.
```

Problem F - Pirate Purchasing

You are the pirate supply officer, PiQueue, on the famous pirate ship, Jaffyboat. Jaffyboat has just returned to port in the small town of Maryburg, and you are responsible for ensuring the ship has the right resources for the next pirate journey. Maryburg likes the pirates of Jaffyboat because, other than an occasional fight at the local saloon, they contribute mightily to the town's economy. The supplies needed for each journey are the normal piraty stuff - Powder Kegs, Peg Legs, Eye Patches, Telescopes, and Cannonballs. As you return to port, the Captain provides you with two items.

- Pieces of gold to make your purchases. The Captain has plenty of gold from the most recent plunder in the Nova Sea.
- A list of quantities of pirate stuff needed for your next pirate journey.

In performing your job, you know the quantities already onboard, you have the list of quantities needed for your next journey as provided by your captain, you have a list of the cost of each item, and you have the amount of gold pieces to spend. You also must sell the excess onboard items when you have more than needed for the next journey. Selling items increases your purchasing gold pieces. You get to keep any gold pieces left over after your purchase so you compute the number of pieces of gold left over.

Input

Line 1 - An integer G - the number of pieces of gold given to you by the Captain
Line 2 - Five integers separated by spaces. The integers are cost of Powder Kegs, cost of Peg Legs, cost of Eye Patches, cost of Telescopes, and cost of Cannonballs.
Line 3 - Five integers separated by spaces. The integers are the numbers of onboard stock for Powder Kegs, Peg Legs, Eye Patches, Telescopes, and Cannonballs
Line 4 - Five integers separated by spaces. The integers are the numbers of stock needed for the upcoming pirate journey for Powder Kegs, Peg Legs, Eye Patches, Telescopes, and Cannonballs

Output

There is one line of output, which will either be an integer that is the number of gold pieces remaining after purchasing the goods or the string `Captain, I need more gold pieces.`

Sample Input 1

```
800
4 3 2 15 7
20 5 10 2 40
50 6 17 3 80
```

Sample Output 1

```
368
```

Sample Input 2

```
200
4 3 2 15 7
```

```
20 5 10 2 40
50 6 17 3 80
```

Sample Output 1

```
Captain, I need more gold pieces.
```

Sample Input 3

```
800
4 3 2 15 7
20 6 10 2 40
50 5 17 3 80
```

Sample Output 1

```
374
```

Problem G - Square Spiral Staircase

You are visiting your child who attends Youemdubya. Your child spends time with you, but they have plans for Saturday evening. This is your first visit to the small town of Maryburg. You decide to stroll around the downtown section of Maryburg while you child pursues their Saturday evening activities. During your strolling, you stumble upon a gorgeous building and enter the front door. Inside you discover a square spiral staircase. There is also a person inside who has climbed the square spiral staircase. The person is obviously tired from climbing the square spiral staircase. In a weak voice the person tells you.

- There is a pot of gold at the top.
- The pirate ship Jaffyboat donated the pot of gold to the town of Maryburg, which is located at the top of the square spiral staircase.
- The armed pirate, PiQueue, is watching the pot of gold.
- Each step has a number.
- The square spiral staircase has a magic spell on it that prevents the use of memory aids such as electronic devices and pencils/paper. You must use your memory mansion to remember the step numbers.
- The armed pirate will give you the pot of gold if you can recite the step numbers in the order you stepped on them.

Input

Line 1 - An integer, N, giving the length of the side of the square spiral staircase. N will be less than or equal to 20.

The next N lines contain N integers separated by spaces, giving the numbers on the steps on the square spiral staircase.

Output

Your program must print the step numbers in the order that you stepped on them. This list of numbers must be on one line and the numbers must be separated by a single space.

Sample Input 1

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

Sample Output 1

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

Sample Input 2

```
4
1 2 3 4
5 7 6 8
9 10 11 12
13 14 15 16
```

Sample Output 2

```
1 2 3 4 8 12 16 15 14 13 9 5 7 6 11 10
```

Problem H - Diamonds in the Rough

Using the gold found at the top of the square spiral staircase, you decide to purchase a diamond for your special friend. The Jewelry Cube in downtown Marybug has many to choose from. The sales person shows you diamonds as a pair of integers. You do not want to examine integers. You want to see each diamond, so you create a program that diagrams the various diamonds using the pair of integers. The diamond tiers are evenly spaced so the diagrams look like gorgeous diamonds.

Input

Two non negative integers on two lines.
Line 1 - the amount of points there will be on each end of the diamond.
Line 2 - the amount of periods on the largest tier of the diamond.

Output

Output is a beautiful Diamond created using periods and spaces.

Sample Input 1

```
1
7
```

Sample Output 1

```
   .
  ...
 .....
.......
 .....
  ...
   .
```

Sample Input 2

```
3
7
```

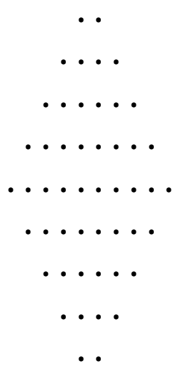
Sample Output 2

```
  ...
 .....
.......
 .....
  ...
```

Sample Input 3

```
2
10
```

Sample Output 3



Problem I - Favorite Integers

The small town of Maryburg is known for its famous triple scoop ice cream cones, which are served on chocolate waffle cones at the Conery Creamery. People come from miles away to purchase and eat a triple scoop ice cream cone. A team of shady looking census workers visited Maryburg for a strangely specific survey. The census workers arrived to collect the favorite integer numbers from the town's population. There are several census workers behind a survey booth. The townsfolk form a line leading to the survey booth. Each person is wearing two COVID19 masks and they are separated by six feet. Families are congregating together in line. We know they are families because each is wearing a hat with the family seal. When a person arrives at the survey booth, the census workers write down their favorite integer on their ledger. If the integer has already been selected by someone, the census workers ask for another favorite number until the person provides a number that is not on the census worker's ledger. The census workers collect a long list of favorite integers, where each integer is unique, and depart with their ledger. Your challenge is to find all triple scoops of integers in the ledger. A triple scoop of integers satisfies the following.

- The first number is less than the second number
- The second number is less than the third number.

Input

Line 1 - an integer number N - the number of test cases.
Each test consists of two lines.
Test Case Line 1 - contains the number of integers on our ledger. This number is less than or equal to 20.
Test Case Line 2 - has the favorite integers separated by spaces, in the order they were transcribed on the list.

Output

N lines where each line contains the number of triple scoops in the test case line 2.

Sample Input

```
3
3
256 1001 1002
4
1 2 3 4
5
12 23 11 25 37
```

Sample Output

```
1
4
5
```

Problem J - Perfect Favorite Integer

The Census central office is in Deecee. When the census workers arrive back at the central office with the Maryburg ledger of favorite integers, the Chief Integer Officer (CIO) tasks the census bureau to determine if the ledger of favorite integers contains any perfect favorite integers. An integer is a perfect number if the sum of all of its divisors less than itself is equal to the number.

Write a program that identifies if an integer is perfect or not.

Input

Line 1 - an integer number N - the number of test cases. N <= 100.
Each test case consists of one line.
Test Case Line 1 - a positive favorite integer between 2 and 10000.

Output

N lines where each line contains all positive divisors less than the favorite integer on test case line 1. The positive divisors must be in an increasing order. Following the positive divisors, the line contains YES if the favorite integer on the input line is a perfect number and NO otherwise.

Sample Input

```
2
6
34
```

Sample Output

```
1 2 3 YES
1 2 17 NO
```

Problem K - Pirate Purchasing Again

You are the pirate supply officer, PiQueue, on the famous pirate ship, Jaffyboat. Jaffyboat has just returned to port in the small town of Maryburg, and you are responsible for ensuring the ship has the right resources for the next pirate journey. Maryburg likes the pirates of Jaffyboat because, other than an occasional fight at the local saloon, they contribute mightily to the town's economy. The supplies needed for each journey are the normal piraty stuff - Powder Kegs, Peg Legs, Eye Patches, Telescopes, and Cannonballs. You know that the Maryburg Eye Patch Store is out of Eye Patches because the population hoarded Eye Patches, along with toilet paper, and Cloroxy wipes due to COVID19. The Captain says we can do without Eye Patches. This means you have to purchase Powder Kegs, Peg Legs, Telescopes, and Cannonballs. The current pirate journey has not been as lucrative as in the past, and the Captain has a limited amount of gold pieces. The captain assigns priorities to the supplies.

- Powder Kegs - have the highest priority
- Cannonballs - have the second highest priority
- Pegs Legs - have the third highest priority
- Telescopes - have the lowest priority

For this problem, the cost for each item is 1 piece of gold. The captain gives you 20 pieces of gold, which you use to purchase items. The available stock is given, and you must purchase items that have the highest priority. For example, if the stores in Maryburg have 9 Powder Kegs, 11 Cannonballs, 8 Peg Legs, and 29 Telescopes, you must use your 20 pieces of gold to purchase 9 Powder Kegs and 11 Cannonballs.

Input

You have 20 pieces of gold. This is known and not part of the input.
Each item costs 1 piece of gold. This is known and not part of the input.
Line 1 - an integer number N - the number of test cases.
Each test case consists of one line.

Test Case Line 1 - Four integers separated by spaces. The integers are the number of Powder Kegs, the number of Cannonballs, the number of Peg Legs, and the number of Telescopes.

Output

There are N lines of output, one line for each test case. Each line will be four integers that are the number of items purchased - Powder Kegs, Cannonballs, Peg Legs, Telescopes.

Sample Input

```
3
9 11 8 29
10 5 4 2
15 3 4 2
```

Sample Output

```
9 11 0 0
10 5 4 1
15 3 2 0
```

Problem L - Encryption Subscription

After the census workers collected favorite integers, the population of Maryburg became paranoid, especially since the census workers would not accept duplicate favorite integers. The population hired some cyber security experts from Youemdubya to create an encryption app that everyone used to send/receive messages on their mobile phones. The population had to subscribe to the encryption app for $10.00 per month, which has helped fund various activities at Youemdubya. This problem provides you with the encryption algorithm. Please do not share this algorithm with the census workers.

- The plain text message consists of all lower case letters and spaces. If the message needs numbers, special characters, etc., they must be spelled in lowercase letters. The following is a sample message exchange.
  ```
  how old is your baby question mark
  my baby is eight months old period
  ```
- A message is blocked into groups of four characters. If a message is not a multiple of four characters, the message is padded with 1, 2, or 3 spaces to make it a multiple of four characters.
- Letters are assigned the number according to their position in the alphabet. a is 0, b is 1, c is 2, and so on.
- For each letter in the message that is a space, replace it with _.

- For each letter in the message that is not an _, add 11 to its numerical representation, perform integer division on this result by 26, and replace the letter with the letter that the remainder maps to. For example, the letter r is in position 17, and it is mapped to the letter c.
- After performing the preceding steps, for each block of four letters, swap the outer pair and swap the inner pair.

Input

The first line contains an integer N - the total number messages to be encrypted.
The remaining lines contain plain text messages to be encrypted. The maximum number of characters on each line is 80.

Output

N lines where each line has the encrypted version of the message.

Sample Input

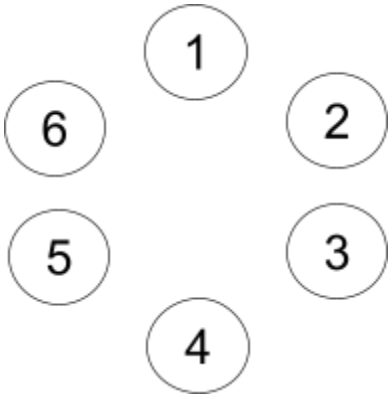
```
3
four
four f
hello
```

Sample Output

```
cfzq
cfzq__q_
wwps___z
```

Problem M - Pass the Potato

When the population in Maryburg is not eating triple scoop ice cream cones, walking the square spiral staircase, purchasing diamonds, sending encrypted messages; they play their favorite game, Pass the Potato. Residents gather together with their favorite sitting chair, form a circle, and pass a potato from one person to another. There are N potato lovers sitting in a circle of N chairs. The following diagram shows a circle of six potato lovers.



To understand the game, consider this sample play, where the exit count is three. The game begins with the person in the first chair holding a potato. The person in chair one begins the count by saying 1, and passes the potato to the person in chair two. The person in chair two counts 2, and passes the potato to the person in chair three. The person in chair three counts 3, which is the exit count. The person in chair three is out of the game. The person in chair three passes the potato to the person in chair four. The person in chair four begins the count by saying 1, and passes the potato to the person in chair five, and the game continues until there is only one person remaining, who is the winner.

In the circle above with an exit count of 3, the first person to be eliminated is in chair three, the second person eliminates is in chair six, the third person eliminated is in chair four, and so on. The winner is the person in chair one.

Input

Line 1 - an integer number N - the number of test cases.
Each test case consists of one line.
Test Case Line 1 - A line with two integers separated by a space. The first integer (1<N<=100000) indicates the number game participants. The second integer is the exit count.

Output

N lines where each line contains a single integer, which is the winner's chair number.

Sample Input

```
2
6 3
6 4
```

Sample Output

```
1
5
```

Problem N - Bamazon Warehouse Packing

During COVID19 the people of Maryburg and all the other cities, towns, and counties avoided shopping in person. As a result the business of the online shopping company - Bamazon - thrived. Bamazon had to hire lots of people to work in their warehouses. In order to retain a high level of packing quality, Bamazon hired you as a quality assurance package checker. As you know, Bamazon packages arrive in boxes, and often a package has a box within a box. In this problem Bamazon packages can have a box in a box in box and so on. Items in a box are the `*` character. Boxes are denoted with `[]` characters. Boxes can hold multiple items.

- A single item in a box is shown as: `[*]`
- Five items in a box is shown as: `[*****]`
- A box with one item in a box with one item is shown as: `[*[*]]` or as: `[[*]*]`

As the quality control package checker, you must ensure that packages have been correctly assembled. The following are samples of correctly assembled packages.

- `[*[***]]`
- `[[[***]*]***]`
- `[*[]]` - Oops. You should have caught this because the inner box is empty, but since you just started and not been properly trained, you mark it as correct. It does have balanced brackets. Balanced, empty boxes are correct for the problem's answers.

The following are samples of incorrectly assembled packages.

- `[[*]`
- `[[[]`

Input

The first line contains an integer N - the total number packages to be verified.
The remaining lines contain packages you must verify. Each will contain 80 or less characters.

Output

N lines where each line has either `Correct` or `Incorrect` that indicates if the package was packed correctly or incorrectly.

Sample Input

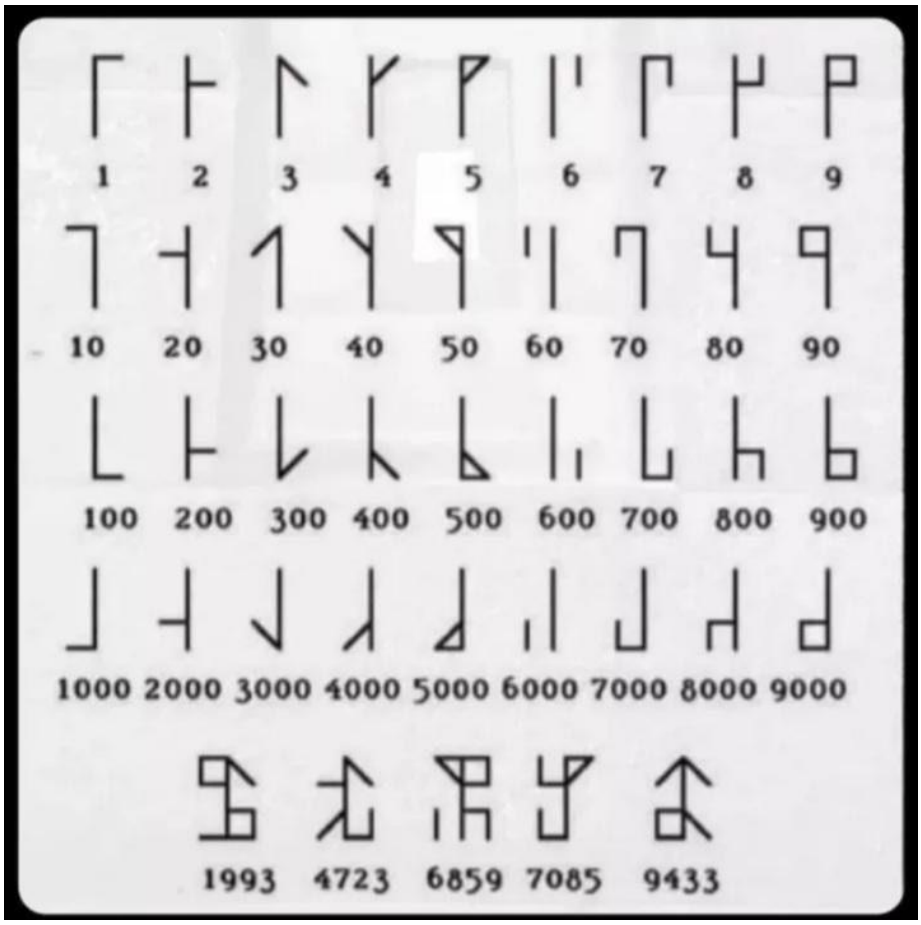
```
11
[*[***]]
[[[***]*]***]
[[*]
[*]
[[[]
][
*[]
[]
[*[*[*]*]*]
[***[***[***[***]***]***]***]
[*[[]]]
```

Sample Output

```
Correct
Correct
Incorrect
Correct
Incorrect
Incorrect
Incorrect
Correct
Correct
Correct
Correct
```

Problem O - Monk EZ

The Cistercian monks invented a numbering system in the 13th century, which meant that any number from 1 to 9999 could be written using a single symbol. Many people believe the Cistercian monks immigrated to Maryburg many years ago and started Youemdubya college, but this has not been confirmed. The monk number symbols were used for years, divisions of tests, the numbering of notes and other lists, indexes and concordances, arguments in easter tables, and even for musical notation. Create a program that translates the numbers we usually use into these symbols. For this problem, each number is a 7 by 7 matrix of periods. You can notice that the middle column of the matrix contains periods. Each wing consists of three periods. If you think of the rows as 0 through 6 and wings beginning to the right/left of the center column, each wing begins in row 0, 2, 4, or 6.



Input

Input is a single, non-negative integer that is the number to be translated into the symbol

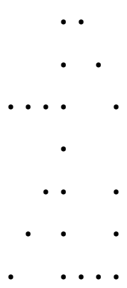
Output

Output is a 7 by 7 output of the translated symbol using "`.`"s

Sample Input 1

```
4723
```

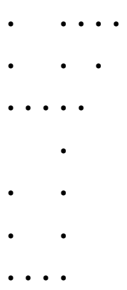
Sample Output 1



Sample Input 2

```
7085
```

Sample Output 2



Contest Input/Output[2]

This guide contains the basic way of doing input and output in Java, Python, and C++. In programming contests, you perform I/O from/to the terminal screen. Do not open any files for input or output. Do not output any prompts. If a problem instructs you to read a line containing a number, just read the number - it will be there. Do not print a prompt like “Enter a number”.

Java Input

Input in Java can be done with the `Scanner` class which must be imported first:
`import java.util.Scanner;`
Then a `Scanner` object must be created:
`java.util.Scanner in = new java.util.Scanner(System.in);`
1. To read in one line of input into a string, use the scanner’s `nextLine` method: `String line = in.nextLine();`
2. To read in a single word into a `String`, stopping at a space, use the `next` method: `String word = in.next();`
3. To read a numerical value use the `nextInt` method for integers, or the `nextDouble` method for real numbers:

```
int number1 = in.nextInt();
double number2 = in.nextDouble();
```

Java Output

1. To output a `String` constant or variable, use the `System.out.println` function which takes one argument, prints it to the screen, then prints a new line:

```
System.out.println("This will be printed") // print a message
int x;
System.out.println(x); // print a value
```

2. To output something without a new line at the end, use the `System.out.print` function which outputs its argument with no newline:

```
// print a message with no new line
System.out.print("The value of X is ")
// print a value with no newline
int x;
System.out.print(x);
```

Python Input

1. The `input` function in Python reads in one line of input and returns it as a string.

[2] Section modified from Dr. Finlayson’s Spring 2018 UMW Contest.

```
line = input() # read in a string
```

2. In order to convert from a string to a number, you can use the int function for integers, or the float function for real numbers:

```
number = int(input()) # read in string and convert to int
```

3. In order to break a line of input into multiple strings, separated by spaces, you can use the `.split()` function which returns a list of strings:

```
line = input() # read in whole line
words = line.split() # split line into strings based on spaces
```

Python Output

1. Output in Python is done with the print function which outputs all of its arguments, separated by spaces, and puts a newline at the end:

```
print("X is equal to", x) # print msg, number, and new line
```

2. In order to prevent print from putting spaces between each item, pass sep=' ' as an arg:

```
print("X is equal to", x, sep= ) # no space between msg and value
```

3. In order to prevent print from putting a new line at the end, pass end=' ' as an argument:

```
print("X is equal to", x, end= ) # no new line added
```

C++ Input

1. To skip over whitespace (spaces or new lines), and read in a single value (such as an integer or string), use `cin >> value;`

```
int number;
cin >> number; // read one integer
char str1[100];
cin >> str1; // read one character string
string str2;
cin >> str2; read one string object
```

2. To read in one entire line of input, which may contain spaces, use `cin.getline`.

```
char str1[100];
cin.getline(str1, 100) // read character string upto 100 chars
string str2;
getline(cin, str2); // read in a string object
```

C++ Output

Output is done with `cout <<` in C++ which can take any built-in data type. For example:

```
int x;
cout << "X is equal to " << x << endl; // print msg, int, new line
```