1. Use a for loop to print a box like the one below. Allow the user to specify how wide and how high the box should be.

```
*******************
*******************
*******************
*******************
```

2. Use a for loop to print a box like the one below. Allow the user to specify how wide and how high the box should be.

```
*******************
*                 *
*                 *
*******************
```

3. Use a for loop to print a triangle like the one below. Allow the user to specify how high the triangle should be.

```
*
**
***
****
```

4. Use a for loop to print an upside down triangle like the one below. Allow the user to specify how high the triangle should be.

```
****
***
**
*
```

5. Let us write a program that generates 10000 random numbers between 1 and 100 and counts how many of them are multiples of 12.

1. A Fibonacci sequence is a sequence of numbers where each successive number is the sum of the previous two. The classic Fibonacci sequence begins: 1, 1, 2, 3, 5, 8, 13,…. Write a program that computes the nth Fibonacci number where n is a value input by the user. For example, if $n = 6$, then the result is 8.

2. Write a program that approximates the value of π by summing the terms of this series: $4/1 - 4/3 + 4/5 - 4/7 + 4/9 - 4/11 + \ldots$ The program should prompt the user for n, the number of terms to sum, and then output the sum of the first n terms of this series. Have your program subtract the approximation from the value of `math.pi` to see how accurate it is.

3. An acronym is a word formed by taking the first letters of the words in a phrase and making a word from them. For example, RAM is an acronym for “random access memory.” Write a program that allows the user to type in a phrase and then outputs the acronym for that phrase. Note: the acronym should be all uppercase, even if the words in the phrase are not capitalized.

4. Numerologists claim to be able to determine a person’s character traits based on the “numeric value” of a name. The value of a name is determined by summing up the values of the letters of the name where ’a’ is 1, ’b’ is 2, ’c’ is 3 etc., up to ’z’ being 26. For example, the name “Zelle” would have the value $26+5+12+12+5 = 60$ (which happens to be a very auspicious number, by the way). Write a program that calculates the numeric value of a single name provided as input.

5. Write a program that counts the number of words in a sentence entered by the user.

6. Write a program that calculates the average word length in a sentence entered by the user.

7. The speeding ticket fine policy in Podunksville is $50 plus $5 for each mph over the limit plus a penalty of $200 for any speed over 90 mph. Write a program that accepts a speed limit and a clocked speed and either prints a message indicating the speed was legal or prints the amount of the fine, if the speed is illegal.

8. A formula for computing Easter in the years 1982–2048, inclusive, is as follows: let $a = year\%19$, $b = year\%4$, $c = year\%7$, $d = (19a + 24)\%30$, $e = (2b + 4c + 6d + 5)\%7$. The date of Easter is March $22 + d + e$ (which could be in April). Write a program that inputs a year, verifies that it is in the proper range, and then prints out the date of Easter that year.

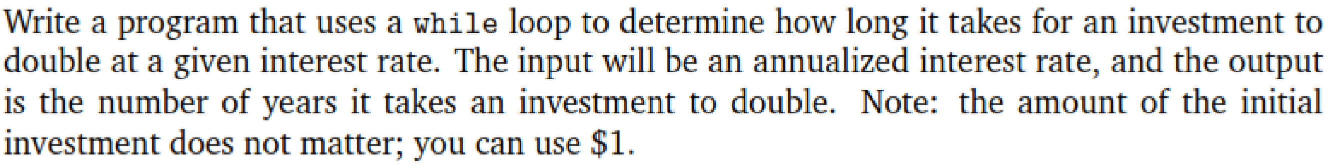
9. Write a program that accepts a date in the form month/day/year and outputs whether or not the date is valid. For example 5/24/1962 is valid, but 9/31/2000 is not. (September has only 30 days.)

10. The days of the year are often numbered from 1 through 365 (or 366). This number can be computed in three steps using int arithmetic:

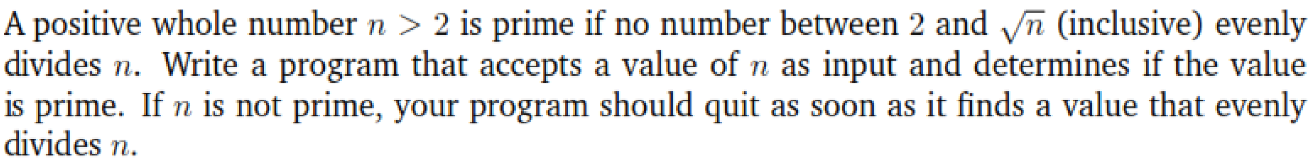
    (a) $dayNum = 31(month - 1) + day$

    (b) if the month is after February subtract $(4month + 23)/10$

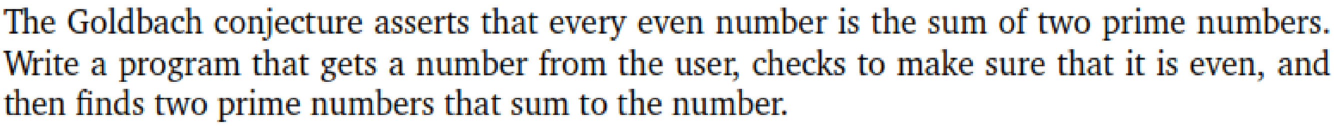
    (c) if it’s a leap year and after February 29, add 1



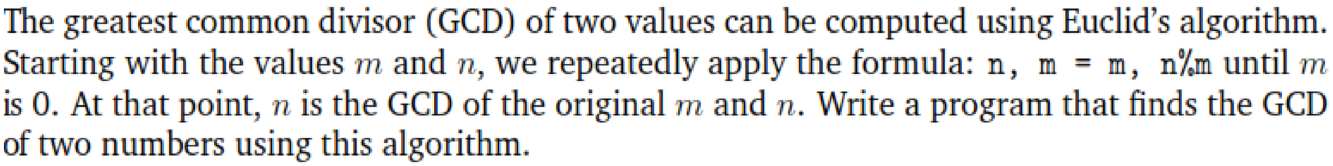
11. Write a program that uses a `while` loop to determine how long it takes for an investment to double at a given interest rate. The input will be an annualized interest rate, and the output is the number of years it takes an investment to double. Note: the amount of the initial investment does not matter; you can use $1.



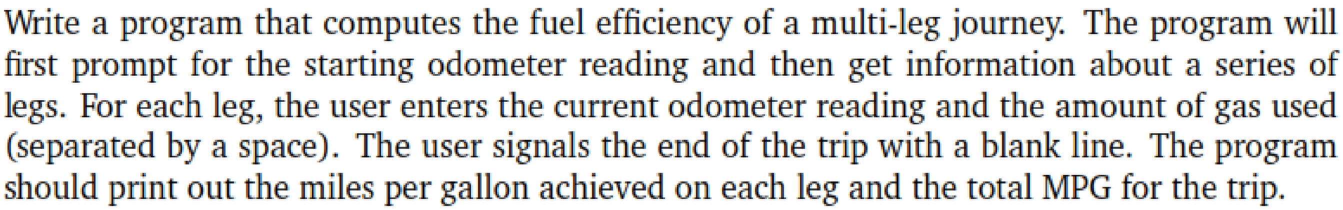
12. A positive whole number $n > 2$ is prime if no number between 2 and $\sqrt{n}$ (inclusive) evenly divides n. Write a program that accepts a value of n as input and determines if the value is prime. If n is not prime, your program should quit as soon as it finds a value that evenly divides n.



13. The Goldbach conjecture asserts that every even number is the sum of two prime numbers. Write a program that gets a number from the user, checks to make sure that it is even, and then finds two prime numbers that sum to the number.



14. The greatest common divisor (GCD) of two values can be computed using Euclid's algorithm. Starting with the values m and n, we repeatedly apply the formula: `n, m = m, n%m` until m is 0. At that point, n is the GCD of the original m and n. Write a program that finds the GCD of two numbers using this algorithm.



15. Write a program that computes the fuel efficiency of a multi-leg journey. The program will first prompt for the starting odometer reading and then get information about a series of legs. For each leg, the user enters the current odometer reading and the amount of gas used (separated by a space). The user signals the end of the trip with a blank line. The program should print out the miles per gallon achieved on each leg and the total MPG for the trip.

16. (a) One way to find out the last digit of a number is to mod the number by 10. Write a program that asks the user to enter a power. Then find the last digit of 2 raised to that power.
(b) One way to find out the last two digits of a number is to mod the number by 100. Write a program that asks the user to enter a power. Then find the last two digits of 2 raised to that power.
(c) Write a program that asks the user to enter a power and how many digits they want. Find the last that many digits of 2 raised to the power the user entered.

17. Write a program that asks the user to enter a number and prints the sum of the divisors of that number.

18. Write a program that asks the user to enter a string. The program should create a new String called new_string from the user's string such that the second character is changed to an asterisk and three exclamation points are attached to the end of the string. Finally, print new_string. Typical output is shown below:

```
Enter your string: Qbert
Q*ert!!!
```

19. Ask the user for a number and then print the following, where the pattern ends at the Number that the user enters.

```
1
   2
     3
       4
```

20. Write a program that asks the user to enter two strings of the same length. The program should then check to see if the strings are of the same length. If they are not, the program should print an appropriate message and exit. If they are of the same length, the program should alternate the characters of the two strings. For example, if the user enters abcde and ABCDE the program should print out AaBbCcDdEe.

21. Write a program that asks the user for an integer and creates a list that consists of the Factors of that integer.

22. Write a program that removes any repeated items from a list so that each item appears at Most once. For instance, the list [1,1,2,3,4,3,0,0] would become [1,2,3,4,0].

23. Write a program to find all numbers between 1 and 1000 that are divisible by 7 and end in A 6.

24. Adding certain numbers to their reversals sometimes produces a palindromic number. For instance, 241 + 142 = 383. Sometimes, we have to repeat the process. For instance, 84 + 48 =132 and 132 + 231 = 363. Write a program that finds both two-digit numbers for which this process must be repeated more than 20 times to obtain a palindromic number.

25. The digital root of a number n is obtained as follows: Add up the digits n to get a new number. Add up the digits of that to get another new number. Keep doing this until you get a number that has only one digit. That number is the digital root.
    For example, if n = 45893, we add up the digits to get 4 + 5 + 8 + 9 + 3 = 29. We then add Up the digits of 29 to get 2 + 9 = 11. We then add up the digits of 11 to get 1 + 1 = 2. Since 2 has only one digit, 2 is our digital root.
    Write a function that returns the digital root of an integer n. [Note: there is a shortcut, where the digital root is equal to n mod 9, but do not use that here.]

26. Write a function called matches that takes two strings as arguments and returns how many matches there are between the strings. A match is where the two strings have the same character at the same index. For instance, 'python' and 'path' match in the first, third, and fourth characters, so the function should return 3.

27. Write a class called Product. The class should have fields called name, amount, and price, holding the product’s name, the number of items of that product in stock, and the regular price of the product. There should be a method get_price that receives the number of items to be bought and returns a the cost of buying that many items, where the regular price is charged for orders of less than 10 items, a 10% discount is applied for orders of between 10 and 99 items, and a 20% discount is applied for orders of 100 or more items. There should also be a method called make_purchase that receives the number of items to be bought and decreases amount by that much.