

Week 4 Assignment

Questions

Question 1

In a particular jurisdiction, taxi fares consist of a base fare of \$4.00, plus \$0.25 for every 140 meters travelled. Write a function that takes the distance travelled (in kilometers) as its only parameter and returns the total fare as its only result. Write a main program that demonstrates the function.

Hint: Taxi fares change over time. Use constants to represent the base fare and the variable portion of the fare so that the program can be updated easily when the rates increase.

Question 2

A prime number is an integer greater than one that is only divisible by one and itself. Write a function that determines whether or not its parameter is prime, returning `True` if it is, and `False` otherwise. Write a main program that reads an integer from the user and displays a message indicating whether or not it is prime. You can use your solution of **Week 3 : Question 4**. Ensure that the main program will not run if the file containing your solution is imported into another program.

Question 3

In this exercise you will create a function named `nextPrime` that finds and returns the first prime number larger than some integer, `n`. The value of `n` will be passed to the function as its only parameter. Include a main program that reads an integer from the user and displays the first prime number larger than the entered value. Import and use your solution to **Question 1** while completing this exercise.

Question 4

If you have 3 straws, possibly of differing lengths, it may or may not be possible to lay them down so that they form a triangle when their ends are touching. For example, if all of the straws have a length of 6 inches then one can easily construct an equilateral triangle using them. However, if one straw is 6 inches long, while the other two are each only 2 inches long, then a triangle cannot be formed. More generally, if any one length is greater than or equal to the sum of the other two then the lengths cannot be used to form a triangle. Otherwise they can form a triangle.

Write a function that determines whether or not three lengths can form a triangle. The function will take 3 parameters and return a Boolean result. If any of the lengths are less than or equal to 0 then your function should return `False`. Otherwise it should determine whether or not the lengths can be used to form a triangle using the method described in the previous paragraph, and return the appropriate result. In addition, write a program that reads 3 lengths from the user and demonstrates the behaviour of your function.

Question 5

Write a function that takes three numbers as parameters, and returns the median value of those parameters as its result. Include a main program that reads three values from the user and displays their median.

Hint: The median value is the middle of the three values when they are sorted into ascending order. It can be found using `if` statements, or with a little bit of mathematical creativity.

Question 6

Write a function that takes two positive integers that represent the numerator and denominator of a fraction as its only parameters. The body of the function should reduce the fraction to lowest terms and then return both the numerator and denominator of the reduced fraction as its result. For example, if the parameters passed to the function are 6 and 63 then the function should return 2 and 21. Include a main program that allows the user to enter a numerator and denominator. Then your program should display the reduced fraction.

Hint: In **Week 3 : Question 3** you wrote a program for computing the greatest common divisor of two positive integers. You may find that code useful when completing this exercise.

Bonus Questions

Bonus Question 1

Write a function that determines how many days there are in a particular month. Your function will take two parameters: The month as an integer between 1 and 12, and the year as a four digit integer. Ensure that your function reports the correct number of days in February for leap years. Include a main program that reads a month and year from the user and displays the number of days in that month. You may find your solution to **Week 2 : Question 4** helpful when solving this problem.

Bonus Question 2

A magic date is a date where the day multiplied by the month is equal to the two digit year. For example, June 10, 1960 is a magic date because June is the sixth month, and 6 times 10 is 60, which is equal to the two digit year. Write a function that determines whether or not a date is a magic date. Use your function to create a main program that finds and displays all of the magic dates in the 20th century. You will probably find your solution to **Bonus Question 1** helpful when completing this exercise.

Bonus Question 3

Many recipe books still use cups, tablespoons and teaspoons to describe the volumes of ingredients used when cooking or baking. While such recipes are easy enough to follow if you have the appropriate measuring cups and spoons, they can be difficult to double, triple or quadruple when cooking Christmas dinner for the entire extended family. For example, a recipe that calls for 4 tablespoons of an ingredient requires 16 tablespoons when quadrupled. However, 16 tablespoons would be better expressed (and easier to measure) as 1 cup.

Write a function that expresses an imperial volume using the largest units possible. The function will take the number of units as its first parameter, and the unit of measure (cup, tablespoon or teaspoon) as its second parameter. It will return a string representing the measure using the largest possible units as its only result. For example, if the function is provided with parameters representing 59 teaspoons then it should return the string "1 cup, 3 tablespoons, 2 teaspoons".

Bonus Question 4

An ordinal date consists of a year and a day within it, both of which are integers. The year can be any year in the Gregorian calendar while the day within the year ranges from one, which represents January 1, through to 365 (or 366 if the year is a leap year) which represents December 31. Ordinal dates are convenient when computing differences between dates that are a specific number of days (rather than months). For example, ordinal dates can be used to easily determine whether a customer is within a 90 day return period, the sell-by date for a food-product based on its production date, and the due date for a baby.

Write a function named `ordinalDate` that takes three integers as parameters. These parameters will be a day, month and year respectively. The function should return the day within the year for that date as its only result. Create a main program that reads a day, month and year from the user and displays the day within the year for that date. Your main program should only run when your file has not been imported into another program.

Bonus Question 5

Create a function that takes an ordinal date, consisting of a year and a day within in that year, as its parameters. The function will return the day and month corresponding to that ordinal date as its results. Ensure that your function handles leap years correctly.

Use your function, as well as the `ordinalDate` function that you wrote previously, to create a program that reads a date from the user. Then your program should report a second date that occurs some number of days later. For example, your program could read the date a product was purchased and then report the last date that it can be returned (based on a return period that is a particular number of days), or your program could compute the due date for a baby based on a gestation period of 280 days. Ensure that your program correctly handles cases where the entered date and the computed date occur in different years.