



Objective:

- Defining the prototypes appropriately.
- Distributing the functionality keeping in view the atomicity, Presentation Layer and Business Layer.
- Using By reference mechanism wherever needed.

You are not allowed to use math library without permission.

Task-1:

Write a C++ function named as 'getRandomAlphaNumeric', which randomly returns an English alphabet or a digit (0 ~ 9). The alphabet can be capital or small which should also be decided randomly.

To accomplish the main task given above: You need to develop the tasks as listed below:

1. **char** getRandomCapitalAlpha()
It returns the capital alphabet randomly.
2. **char** getRandomSmallAlpha()
It returns the small alphabet randomly.
3. **char** getRandomDigit()
It returns the random digit randomly.
4. **char** getRandomAlphaNumeric()
Use the above three functions in this function to return the alphanumeric symbol randomly.

Task-2:

Many advanced calculators have a fraction feature that will simplify fractions for you. You are to write a program that will input a positive integer as a numerator and a positive integer as a denominator, and output the fraction in simplest form. That is, the fraction cannot be reduced any further, and the numerator will be less than the denominator in output.

Sample Input/output

	Input	Output
1	Numerator: 28 Denominator: 7	4
2	Numerator: 13 Denominator: 5	2 3/5
3	Numerator: 0 Denominator: 7	0
5	Numerator: 12 Denominator: 0	Infinity

To accomplish the above task, you need to develop the tasks in following order.

1. **int** GCD (**int**, **int**);
It returns the greatest common divisor of the two received integers.
2. **void** reduceFraction (**int** *, **int** *);
It receives two integers by reference and change the numbers in reduced form.
First Parameter: Numerator, Second Parameter: Denominator
For Example: 10/5 will become 2/1, 56/100 will become 14/25
3. **void** printSimplifiedFraction (**int**, **int**);
It uses the above two functions to achieve the original task given in question.
First Parameter: Numerator, Second Parameter: Denominator



Task-3:

Write a program which takes date (year, month, day) from user and displays on console the next date.

Note: A leap year is either divisible by 4 yet not by 100, or it is divisible by 400.

Input Validation: If user enters wrong/invalid date then you must report user about it.

Leap Year ☺:

Following years are not leap years:

1700, 1800, 1900, 2100, 2200, 2300, 2500, 2600

It is because they are evenly divisible by 100 but not by 400.

To accomplish the main task given above: You need to structure your tasks as follows:

1. **bool** isLeapYear (**int**);
It returns true if the received year is leap otherwise returns false.
2. **void** getNextDate (**int** *, **int** *, **int** *);
It receives three integers (day, month, year) by reference and change the received integers to next date.
3. **void** nextDateApp ();
This function handles the interface of the program and takes input for date and use the above two function to achieve/display the required output on console.

Task-4:

A right triangle can have sides that are all integers. A set of three integer values for the sides of a right triangle is called a Pythagorean triple. These three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse. Find and display all primitive Pythagorean triples for side1, side2 and hypotenuse all no larger than 500.

Example:

There are 16 primitive Pythagorean triples with $c \leq 100$:

(3 , 4 , 5)	(5, 12, 13)	(7, 24, 25)	(8, 15, 17)
(9, 40, 41)	(11, 60, 61)	(12, 35, 37)	(13, 84, 85)
(16, 63, 65)	(20, 21, 29)	(28, 45, 53)	(33, 56, 65)
(36, 77, 85)	(39, 80, 89)	(48, 55, 73)	(65, 72, 97)

A Pythagorean triplet is a set of three natural numbers, $a < b < c$, for which,
$$a^2 + b^2 = c^2$$

For example, $3^2 + 4^2 = 9 + 16 = 25 = 5^2$.

A primitive Pythagorean triple is a Pythagorean triple (a, b, c) such that $\text{GCD}(a, b, c) = 1$

To accomplish the main task given above: You need to structure your tasks as follows:

1. **int** GCD (**int**, **int**);
It returns the greatest common divisor of the two received integers.
2. **bool** isPythagoreanTriplet (**int** , **int** , **int**);
It returns true if the received integers (a, b, c) as sides of triangle are Pythagorean triplet otherwise return false.
3. **void** printPythagoreanTriplet (**int**=500);
This function prints all the Pythagorean triplet up to 500 and uses the above two functions to achieve the original task.
The functions receive the limit up to which triplets needs to be checked which is by default 500.



Task-5:

Write a function, which receives following three arguments from the user:

- ❖ Number
- ❖ Source base
- ❖ Target base

It then converts the number from source base to targeted base and returns the resultant number in target base.

Note: You can assume that the base will be less than or equal to 10. The default source base will be decimal and default target base will be binary.

Structure your code wisely keeping in view the structuring that is given in Tasks 1~4.