



Department of Computer Science
University of Engineering and Technology, Lahore



Write the C++ functions of the following problems.

Problem 1:

Try finding your ancestors and offspring with code.

Create a function that takes a number x and a character y ('m' for male, 'f' for female), and returns the name of an ancestor (m/f) or descendant (m/f).

- If the number is negative, return the related ancestor.
- If positive, return the related descendant.
- You are generation 0. In the case of 0 (male or female), return "me!".

Generation	Male	Female
-3	great grandfather	great grandmother
-2	grandfather	grandmother
-1	father	mother
0	me!	me!
1	son	daughter
2	grandson	granddaughter
3	great grandson	great granddaughter

Test Cases:

- `generation(2, 'f') → "granddaughter"`
- `generation(-3, 'm') → "great grandfather"`
- `generation(1, 'f') → "daughter"`

Problem 2:

Create a function that takes the length, width, height (in meters) and output unit in which you want to see the answer and returns the volume of a pyramid in the correct unit.

Notes:

- The units used are limited to: millimeters, centimeters, meters and kilometers.
- Ensure you return the answer and add the correct unit in the format cubic <unit>.

Test Cases:

- `pyramidVolume(4, 6, 20, "centimeters")` → "160000000.000 cubic centimeters"
- `pyramidVolume(1843, 1823, 923, "kilometers")` → "1.034 cubic kilometers"
- `pyramidVolume(18, 412, 93, "millimeters")` → "229896000000000.000 cubic millimeters"

Problem 3:

A leap year has one day added to February for being synchronized with the seasonal year. A leap year appears with a regular frequency, and You can check whether a year is leap or not by using this mathematical logic:

Leap Year:

- If a year is divisible by 4, 100 and 400 then it is a leap year.
- If a year is divisible by 4 but not divisible by 100 then it is a leap year

Not a Leap Year:

- If a year is not divisible by 4 then it is not a leap year
- If a year is divisible by 4 and 100 but not divisible by 400 then it is not a leap year

Given a year you must implement a function that returns true if it's a leap year, or false if it's not.

Test Cases:

- `isLeap(2020)` → true
// Exactly divided by 4 and not by 100.
- `isLeap(1800)` → false
// Exactly divided by 4, and exactly divided by 100, but not 400.
- `isLeap(2000)` → true
// Exactly divided by 4, 100 and 400.
- `isLeap(2019)` → false
// It can't be exactly divided by 400 or by 4.

Problem 4:

Create a function that takes 2 years as input, the difference between the years will be exactly 10. For example if the first input is the year 2005 then the next input will be 2015. Your goal is to count how many leap years there were between these 10 years and return that count from the function.

Test Cases:

countLeapYears(2010, 2020) → 2

countLeapYears(2016, 2026) → 3

Hint:

It's not always "**Hard work is the key to success**". It's in our case "**Smart Work is the key to success**". You can use the previous isLeap function to solve the problem 4.

Problem 5:

Mubashir has a cat and a dog. He purchased both of them at the same time humanYears ago. Create a function which takes an argument of humanYears and displays (cout) humanYears, catYears, and dogYears.

Human Years

- Human Years >= 1
- Human Years are whole numbers only.

Cat Years

- 15 cat years for the first year.
- +9 cat years for second year.
- +4 cat years for each year after that.

Dog Years

- 15 dog years for first year
- +9 dog years for second year
- +5 dog years for each year after that

Test Cases:

calculateYears(1) → Human Years: 1
Cat Years: 15
Dog Years: 15

calculateYears(2) → Human Years: 2
Cat Years: 24
Dog Years: 24

calculateYears(10) → Human Years: 10
Cat Years: 56
Dog Years: 64