

# PROBLEMS THAT INTRODUCE YOU TO NEW PROGRAMMING CONCEPTS

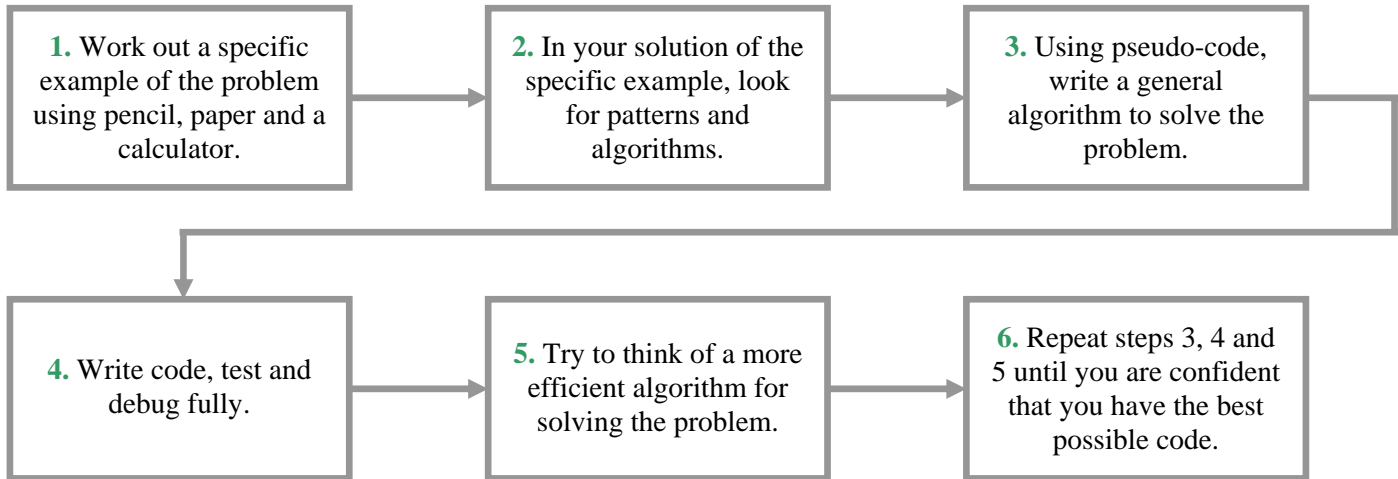
## Introduction

The main purpose of these practice problems is to

- **Help you understand the importance of analyzing problems and planning solutions BEFORE YOU ATTEMPTING TO WRITE PROGRAMS TO SOLVE THE PROBLEMS!** Remember that a program is a set of instructions that a computer follows to solve a problem. If you do not know how to solve a particular problem, it is impossible for you to write a program that solves that problem!
- **Introduce you to new programming concepts.**
- **Help you to develop the facility for quickly choosing an appropriate programming structure.**
- **Build a library of frequently used code.**

For instance, you will learn to decide quickly whether to choose a “**Do ... Loop Until**,” a “**Do ... Loop While**,” a “**Do Until ... Loop**,” a “**Do While ... Loop**” or a “**For ... Next**” structure. Furthermore, many of the problems presented below occur frequently while developing solutions to larger problems. For example, the problem of generating random numbers without repetition is very often encountered in many different programming situations.

When working on these problems, keep in mind the following problem solving model:



## A Summary of What you need to understand to solve the Problems on the Next Page

- One-Dimensional Arrays
- “**Do ... Loop Until**,” “**Do ... Loop While**,” “**Do Until ... Loop**,” “**Do While ... Loop**” and “**For ... Next**” loops
- String Manipulation Techniques and Functions
- The **Mod** Operator
- The “\” Operator
- How to Generate Random Integers
- How to Sort Numbers from Smallest to Largest
- The Binary Representation of Positive Integers and the Twos Complement Binary Representation of Negative Integers
- How to Dynamically Resize an Array (Look up “**ReDim**” in MSDN Help)

### Mathematical Practice Problems

1. A number is called *perfect* if the sum of its *proper divisors* is equal to the number itself. Two examples of perfect numbers are 6 and 28 because  $6 = 1 + 2 + 3$  and  $28 = 1 + 2 + 4 + 7 + 14$ .  
Write a program that finds *all* perfect numbers within the range of an *Integer* variable.
2. Write a program that finds the *greatest common divisor* of any two integers. For example, the greatest common divisor (GCD) of 24 and 40 is 8.
3. Write a program that finds the *least common multiple* of any two integers. For example, the least common multiple (LCM) of 24 and 40 is 120.
4. Write a program that can add, subtract, multiply and divide two fractions and display the result in lowest terms.
5. Write a program that can factor a *simple trinomial*, if it is factorable. If the simple trinomial is not factorable, the program should display a suitable message
6. Write a program that will convert any *Long* integer specified in *decimal* (base ten) form to *binary* (base 2) form. Note that to convert *negative* integers to binary form, you must understand something called the “twos complement” binary representation of negative integers. You will receive extra credit for learning about the “twos complement” representation without the teacher’s help.
7. The numbers 220 and 284 are called an *amicable pair* because the sum of the proper divisors of 220 is 284 and the sum of the proper divisors of 284 is 220. Write a program that finds *all* amicable pairs within the range of an *Integer* variable.

### Array Practice Problems

1. Fill an array of ten elements *at random* with the numbers 0 to 9 with *no repetition*.
2. Fill an array of ten elements *at random* with *ten different* two-digit numbers.
3. Fill an array of ten elements *at random* with the numbers 1 to 5 so that each number appears *exactly twice*.
4. Display all possible ways to choose a pair of numbers from a list of ten numbers. Note that order does not matter in this problem. For example, the pair “2, 5” is considered the same as the pair “5, 2.”
5. Remove all even numbers from an array of numbers.
6. Split an array of even numbers into two arrays, one of *even* numbers and one of *odd* numbers.
7. Remove any repeated numbers from an array without leaving any gaps.
8. Determine the *mean* (average) of an array of numbers.
9. Determine which number(s) occur most frequently in an array of numbers (i.e. find the *mode*).
10. Determine the *median* of an array of numbers. That is, find the number that divides the array into two halves, one half consisting of all the numbers *less than* the median and the other half consisting of all the numbers *greater than* the median.
11. Remove all the spaces from a string.
12. Remove all the vowels from a word.
13. Scramble the letters in a word.

## General Programming Problems

For each of the exercises given below, develop a Visual Basic program that is efficient, well coded, well documented and visually appealing. Do not forget to write a pseudo-code algorithm to solve each problem *before* you blindly and slavishly start typing Visual Basic code.

1. Write a program that will take as input any value in dollars and cents. It will then calculate how many pennies, nickels, dimes, quarters, “loonies,” “twonies” and paper bills (\$5, \$10, \$20, \$50, \$100 and \$1000) would be needed to equal the entered amount. To make your program visually appealing, search the Internet for images of Canadian coins and paper bills and include them in your user interface.
2. Write a program that takes as input a value in seconds and then converts the value to an equivalent number of hours minutes and seconds (“time converter”).
3. Write a program that conjugates any regular French verb. If you need to brush up on your French grammar, here are the correct endings for the three types of regular verb infinitives:

| -er verbs | -re verbs | -ir verbs |
|-----------|-----------|-----------|
| e         | s         | is        |
| es        | s         | is        |
| e         | -         | it        |

4. Write a program that randomly generates a set of six integers ranging from 1 to 49 (for Lotto 6/49). The numbers should be displayed in ascending order (smallest to largest). Do not forget to use the Internet to search for pertinent graphics!