AUBURN UNIVERSITY

CPSC 3303 Programming Assignment 1

Instructions: For this assignment, you must write three programs in the high-level language you learned in CPSC 1213. For all these exercises, you cannot use built-in functions that perform these operations. In case of a doubt, check with your instructor.

Objectives of this assignment:

- To "dust off" your programming skills in the high-level language you learned in CPSC 1213
- To stress the fact that the variables ultimately are 0s and 1s
- To practice conversion from a base (decimal, hexadecimal, or binary) to another base
- To distinguish between "numbers" and characters

What you need to do:

Programming Exercise 1 (12 points):

Write a program that prompts the user to enter a character *c* that represents a binary digit (a bit!). (Recall that *c* can be only "0" or "1.") Your program must use the *character* type for the input. If the user enters a character "x" that is *not* a bit, you must print out the following error message: "The character x is invalid: x is not a bit." If the character *c* is a bit, your main program must print out its *value* in decimal.

Example 1: If the user enters the character "0," your program must print out the value 0.
Example 2: If the user enters the character "1," your program must print out the value 1.
Example 3: If the user enters the character "B," your program must print out the following error message: "The character B is invalid: B is not a bit."

Programming Exercise 2 (12 points):

Write a program that prompts the user to enter a character *c* that represents a hexadecimal digit (recall that *c* can be "0," "1," "2," …., "8," "9," "A," "B," "C," "D," "E," or "F"). Your program must use the *character* type for the input. If the user enters a character "x" that is *not* a hexadecimal digit, you must print out the following error message: "The character x is invalid: x is not a hexadecimal digit." If the character *c* is a hexadecimal digit, your main program must print out its *value* in decimal.

Example 1: If the user enters the character "4," your program must print out the value 4.
Example 2: If the user enters the character "B," your program must print out the value 11.
Example 3: If the user enters the character "E," your program must print out the value 14.
Example 4: If the user enters the character "I," your program must print out the following error message: "The character I is invalid: I is not a hexadecimal digit."

Programming Exercise 3 (16 points):

Write a program that prompts the user to enter a positive integer *n* (0 up to 2^{32}-1). You must write a function that takes as input *n* and returns a string *s* representing the number *n* in binary. For this assignment, you *must use the method of successive division by 2* to convert the number to binary. Your main program must print out *s*.

Example: If the user enters the number 66, your program must print out 1000010.

AUBURN UNIVERSITY

CPSC 3303 Programming Assignment 1

What you need to turn in:

- You will need to include an electronic copy of your report (standalone) and source code (zipped) of your programs. All programming files (source code) must be put in a zipped folder named "lab1-name," where "name" is your last name. Submit the zipped folder on the assignment page in Canvas; submit the report separately (not inside the zipped folder) as a Microsoft Word or PDF file.
    - You should adhere to the following guidelines:
        - State whether your code works.
        - Clearly explain how to compile and execute your code.
        - If needed/applicable, report/analyze (as appropriate) the results. The quality of analysis and writing is critical to your grade.
        - Good writing and presentation are expected.

How this assignment will be graded:

The program compiles and executes correctly without any apparent bugs, and the code is well-designed and commented	100% Credit
The program compiles and executes with minor bugs, and the code is well-designed and commented	90% credit
The program compiles and executes with major bugs	40% credit
The program compiles but does not produce any meaningful output	5% credit

If the instructor needs additional communications/actions to compile and execute your code, then a 30% penalty will be applied. If the turn-in instructions are not correctly followed, 10 points will be deducted.