ICS 212 Homework #2

In this assignment, extend the gumball machine program which you began in homework #1.

In module GumballMachine.c, create a function called OperateGumballMachine(). Call this function from main(). Note: this is normally how C programs are written (well, "good" programs, anyway); normally very little is done inside main(), except to call functions which do the actual work of the program. Create a loop in this function; which type of loop you use is your choice, but the termination of the loop should be determined by the user's request. Within this loop, ask the user to insert coins, one at a time. The user may insert: nickels, dimes, or quarters. When the money reaches 25c, the coin insertion process stops. Any money in excess of 25c should be returned to the customer, and function DeliverGumBall() (see below) should be called.

● Please use the following standardized input: 'n', 'd', 'q', and 'x' for nickels, dimes, quarters, and exit, respectively. Your program should print instructions on startup, and an error message should the user provide incorrect input.

● For the purposes of this assignment, you may use function rand() supplied by the standard library. Remember to seed the random number generator once (and only once) per program execution using function srand(). Most people seed the random number generator with the time() function.

● In GumBallMachine.c, write a function called DeliverGumBall() which you should call from OperateGumballMachine(). This function should take no parameters; remember to put void between the parentheses to indicate this. For this one assignment, use a global variable in GumBallMachine.c to describe the gumball machine (i.e. the thing which indicates how many gumballs of each color there are). (Ordinarily, such use of globals is verboten, but we haven't discussed the preferred approach in class, yet; we will cover this, shortly). In DeliverGumBall(), a random number should be generated, to determine what color of gumball will be delivered to the customer. You should use a switch statement to take the appropriate action based on the number (the appropriate action in this case, is just to print out the color of gumball, and decrement the appropriate field in the gumball machine struct). Remember to use constants (i.e. using #define) and not magic numbers in your switch. Also remember to supply your switch with a default case.

● In this assignment you should initialize the gumball machine with a different number of gumballs of each color. Thus, you'll have to take care of the special case in which your gumball machine runs out of gumballs of one or more colors. (I don't want to influence you, but one approach might be to use statistical probabilities based on the number of gumballs of each color remaining, to decide what color of gumball to deliver in the first place).

● You should write a small function in GumballMachine.c called RefillGumballMachine() to refill the gumball machine when it is completely empty.

Have fun!

● When you are done, submit your source code as you did for homework 1: upload a ZIP file named *username*A02.zip to Tamarin.