# Programming Assignment 2

1. Ask the user to enter their age and store it in the variable **age.** Write code that will print to the console whether or not a person is allowed into a bar (A person is allowed into a bar if they are 21 years or older)

```
age = <Some value>
#Your code here
```

2. You are driving a little too fast, and a police officer stops you. Ask the driver how fast he was going and store in the variable **speed**. Write code to print to the console the size of your ticket as follows:

   If speed is 60 or less, the result is no ticket.
   If speed is between 61 and 80 inclusive, the result is a small ticket.
   If speed is 81 or more, the result is a big ticket.
   Unless it is your birthday -- on that day, your speed can be 5 higher in all cases. Ask the driver if it is their birthday and store in the variable **bday**.

3. Create a loop that will prompt the user to input a valid number, i.e. non-negative integer, and store it in the variable multNum. You do not need to handle the case where the user inputs a float or string, only positive/negative int values.
   Print “True” to the console if the given non-negative number is a multiple of 3 or 5, but not both. Use the % "mod" operator

```
number = <some value>
#Your code here
```

4. Ask the user to enter three ints, **a b c**, print true if they are in strictly increasing order. This means that the b is greater than a and c is greater than b (a < b < c). For example 2 5 11, or 5 6 7 are strictly increasing but not 6 5 7 or 5 5 7.

5. Write a program that prints the “99 bottles of beer on the wall” song:
   http://www.99-bottles-of-beer.net/lyrics.html

6. The value of **PI** can be determined by the series equation:

$$\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \frac{1}{13} - \cdots)$$

   Write a program to approximate the value of p using the formula given including terms up through 1/99.

7. You are given three numbers by the user, **k**, **small** and **large**. You want to make a row of bricks that is k inches long with the given number of small (1 inch) and large bricks (5 inches). Print “Possible” if it is possible to make the goal by choosing from the given bricks. For example if k = 8, small = 2, large = 3 then it’s not possible because there is no way to get 8 inches out of the given bricks. While k = 7 with small = 2 and large = 3 is possible (1 large and 2 small gives you 7).

```
k = <some value>
smallBrick = <some value>
largeBrick = <some value>
#Your code here
```

8. Write a program that asks for an amount, store it in the variable **change**, and determines the least number of dollar bills and coins needed for change. Note that dollar bills come in : 1, 5, 10, 20, 50 and 100 bills. Coins can be: 1 cent, 5 cents, 10 cents and 25 cents (penny, nickel, dime, quarter).

   Example:

   If the dollar amount is 35.63, then the least number of bills and coins is: 1 x $20 bill, 1 x $10 bill, 1 x $5 bill, 2 quarters, 1 dime, 3 pennies.