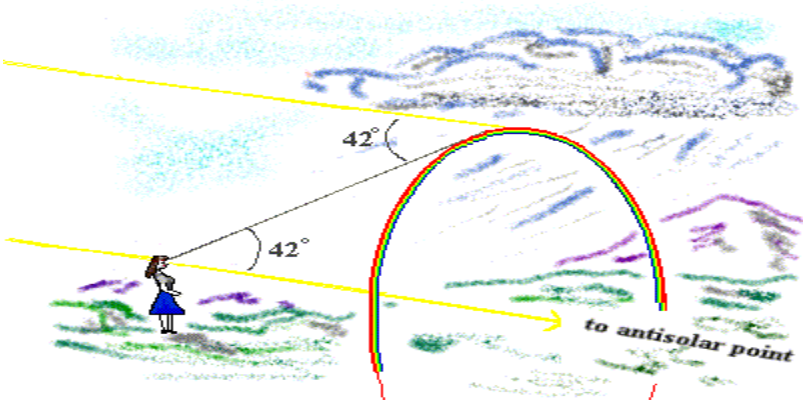
ITSE 2417- Fall 2016
Program Set #1

See Java Grading /Program Guide Sheet for directions and grading/submission information.

1. A rainbow is an optical illusion can be observed whenever there are water drops in the air and sunlight shining from behind the observer at a low altitude angle. Write a Java program given the height of the user (in both feet and inches) and the distance to the rainbow, output the height of the given rainbow to 4 decimal places. Some needed constants to use:




```
REFRACTION_ANGLE = 42.333333;    //measured in degrees
INCHES_PER_FOOT = 12.0;
RADIANS_PER_DEGREE = PI / 180.0;
```

Assume valid input. Output should look similar to below.

Sample Run:

```
Rainbow program:

Please enter the height in feet and inches: 6 3
Your height is 6.25 feet.

Please enter your distance to the rainbow in feet: 50
Your distance to the rainbow is 50.0 feet.

The height of the rainbow is 51.7997 feet.
```

Name the program: `RainbowHeightXX.java`, where XX are your initials.

2. In mathematics, two prime numbers are considered sexy if they have a difference of 6 ("sex" is the Latin word for "six."). In other words, pairs of primes of the form `(p, p + 6)`. Given two integer numbers, write a Java program to determine if the numbers are both prime and have a difference of 6. Input the smaller number fist. Determine if the pair of numbers are sexy or not. Output should look

similar to below. Check for positive values `1-3000`. Do not list out all the sexy numbers- the values must be algorithmically calculated. Output should look similar to below.

Sample Runs:

```
Enter two numbers (1-3000):  2 8
The numbers are not sexy primes.

Enter two numbers (1-3000):  641 647
The numbers are sexy primes.
```

Name the program: `SexyNumbers.java`, where `XX` are your initials.

3. In music, songs can be cumulative which have a simple verse structure that when modified each verse is longer than the verse before it. One such cumulative song is *Old MacDonald had a Farm*. Write a Java program which will generate the lyrics to version of Old MacDonald such that the animals, sounds and order can be changed. Here are the first few lyrics of the song:

```
Old MacDonald had a farm, E-I-E-I-O,
And on that farm he had a pig, E-I-E-I-O,
With an oink-oink here and an oink-oink there,
Here an oink, there an oink, everywhere an oink-oink,
Old MacDonald had a farm, E-I-E-I-O.

Old MacDonald had a farm, E-I-E-I-O,
And on that farm he had a cow, E-I-E-I-O,
With a moo-moo here and a moo-moo there,
Here a moo, there a moo, everywhere a moo-moo,
With an oink-oink here and an oink-oink there,
Here an oink, there an oink, everywhere an oink-oink,
Old MacDonald had a farm, E-I-E-I-O.

Old MacDonald had a farm, E-I-E-I-O,
And on that farm he had a horse, E-I-E-I-O,
With a neigh-neigh here and a neigh-neigh there,
Here a neigh, there a neigh, everywhere a neigh-neigh,
With a moo-moo here and a moo-moo there,
Here a moo, there a moo, everywhere a moo-moo,
With an oink-oink here and an oink-oink there,
Here an oink, there an oink, everywhere an oink-oink,
Old MacDonald had a farm, E-I-E-I-O.
```

Note with *Old MacDonald* the song starts and ends with a common refrain ("`Old MacDonald had a farm, E-I-E-I-O`"). After the first refrain, each verse introduces a new animal from the farm, providing the name of the animal in a lyric context ("`And on that farm he had a pig, E-I-E-I-O`"). Then the sound of the animal is noted in a rhythmic fashion, including sometimes "doubling" the sound:

```
With an oink-oink here and an oink-oink there,
Here an oink, there an oink, everywhere an oink-oink
```

This is followed by the closing refrain. With each subsequent verse a new animal and sound are introduced. Also, note the length of the verses also grows, because after the new animal and its sound are introduced, we also repeat the sound portion for every animal already sung about, in the reverse of the order in which they were introduced. Thus in verse one, we sing about the first animal by name and sound; in verse two, we sing about the second animal by name and sound, but repeat the sound portion for the first animal; and in each subsequent verse we add a new animal, but repeat the sound portions for all previous animals. Continue the *Old MacDonald* song with the following animal/sound replacements, in order:

```
sheep- baa, baa
pig- oink, oink
cow- moo, moo
horse- neigh, neigh
sheep- baa, baa
goose- honk, honk
dog- bow-wow, bow-wow
```

Use a method `buildVerse()`which has two parameters- an animal and a sound for each verse and repeated text. Do not use any `String` class methods or simply write all `Console.WriteLine()` statements that outputs each line of the song in order. Output should look similar to below.

Sample Run (partial):

```
The lyrics to "Old MacDonald had a Farm":

Old MacDonald had a farm, E-I-E-I-O,
And on that farm he had a pig, E-I-E-I-O,
With an oink-oink here and an oink-oink there,
Here an oink, there an oink, everywhere an oink-oink,
Old MacDonald had a farm, E-I-E-I-O.

Old MacDonald had a farm, E-I-E-I-O,
And on that farm he had a cow, E-I-E-I-O,
With a moo-moo here and a moo-moo there,
Here a moo, there a moo, everywhere a moo-moo,
With an oink-oink here and an oink-oink there,
Here an oink, there an oink, everywhere an oink-oink,
Old MacDonald had a farm, E-I-E-I-O.
```

... Rest of the song with animals and sounds

Name the program: `OldMacDonaldXX.java`, where XX are your initials.

4 (**). Holidays can fall on the same day every year such as Christmas Day (December 25th). Not all holidays, however, fall on the same day every year. These floating holidays do have a pattern to their scheduling. The following are six examples of floating holidays:

M.L. King Day	3rd Monday of January
Presidents Day	3rd Monday of February

Memorial Day	Last Monday of May
Labor Day	1st Monday of September
Election Day	Tuesday after the 1st Monday of November
Thanksgiving Day	4th Thursday of November

Write a Java program to determine the actual date each of the following six holidays fall for a given year. Input will be a year from 1700-2100. For each year, output each of the dates of the six floating holidays celebrated that year, and in the order listed above. Express the dates in a `mm/dd/yyyy` format. Output should look similar to below.

Sample Run:

```
Enter a year (1700-2200):  2057

M L King Day        01/15/2057
Presidents Day      02/19/2057
Memorial Day        05/28/2057
Labor Day           09/03/2057
Election Day        11/06/2057
Thanksgiving Day    11/22/2057
```

Name the program: `FloatingHolidays.java`, where XX are your initials.