Grade 11 Review Part II – Control Flow, Loops

Use only if statements and loops. For now, in NetBeans, it's easiest to make a separate project for each.

1. `SortThreeLetters.java` Write a program that prompts for three lower case letters separately (`String`). Assuming all inputs are valid, sort and output the three letters in alphabetical order, otherwise, output an error message. No arrays for now.
   Sample run:

```
Enter a single letter followed by 'Enter':
i
c
s

Your letters in alphabetical order: c i s
```

2. `LeastPowerOfTwo.java` Write a program that reads a positive integer and then finds and prints the smallest power of two that is greater than or equal to the number that was given. For example, if the program reads the value 25, it should note that $32 = 2^5$ is the smallest power of two greater than or equal to 25.
   Sample run:

```
Enter a positive integer: 25
32 is the smallest power of 2 greater than 25.
```

3. `LeapYear.java` A leap year is one with 366 days instead of 365. Write a program to determine whether a particular year is a leap year. Recall that a year is a leap year if it is divisible by 4. But if the year is divisible by 100, it is only a leap year if it is also divisible by 400 (why?!). Ask the user for a year and print the result.
   Sample run:

```
Enter a year: 2000
The year 2000 is a leap year.

Enter a year: 2018
The year 2018 is not a leap year.

Enter a year: 2100
The year 2100 is not a leap year.
```

4. `PythagoreanTriple.java` Three positive integers a, b, and c with $a < b < c$ form a Pythagorean Triple if $a^2 + b^2 = c^2$. For example (3, 4, 5) is a Pythagorean Triple since $3^2 + 4^2 = 5^2$. Write a program that first prompts the user for a positive integer and then finds and prints all Pythagorean Triples whose largest member is less than or equal to that integer. [Challenge: do not print Pythagorean Triples that are multiples of other triples, e.g. (6, 8, 10) is simply 2 × (3, 4, 5).]
Sample run:

```
Enter a positive integer: 14
Pythagorean Triples:
(3, 4, 5)
(6, 8, 10)
(5, 12, 13)
```