COP-2250 - Java Programming
Professor: Michael Robinson
e-mail   : michael.robinson@fiu.edu
Web Page : www.cs.fiu.edu/~mrobi002/teaching

Program 2aFLAME

Make sure your program is properly documented and aligned uniformally, looking professionally,
I will take points off if it not.

Purpose: Implement methods, if, else if, else, for, print, println, printf,
primitive types, String class with several methods, arrays, swap.

1 - In the main method (Worth 1 point)
- Create the following variables:
```
int i = 7;
int j = 9;
double x = 72.5;
double y = -0.34;
```

- Passing the previous variables, call the following methods:
```
processAbsoluteValues
processRoundValues
processCeilingValues
processFlooringValues
processLogValues
```

- Create a single dimension array containing ONE index,
which will have ONE string with your full name as follows:

"first name @ middle name ^ last name and ONE space"
i.e. "George & Washington ^ Rodriguez "
Note: if you do not have a middle name use MN.

2 - (Worth 1 point) Create the previous methods, and in their corresponding methods,
using the data received implement the following Math methods:
abs, round, ceil, floor, log

3 - Create a method called myName (Worth 2 points).
Using the for loop, if, else if, else, ||, &&, printf, and any other
command you want that we have learned in class,
- Examine each character of your name and determine if it is:
  - a space, a vowel (a,e,i,o,u,A,E,I,O,U), a consonant (b,w,g, etc), the symbol ^
    or the symbol @, and using the current value of x in your for loop print:

```
character [x] located at position x is a consonant   or
character [x] located at position x is a vowel       or
character [x] located at position x is a space       or
character [x] located at position x is a symbol

ie: My name is [George @ Washington ^ Rodriguez]

character [G] located at position 0 is a consonant
character [e] located at position 1 is a vowel
character [o] located at position 2 is a vowel
character [r] located at position 3 is a consonant
character [g] located at position 4 is a consonant
character [e] located at position 5 is a vowel
character [ ] located at position 6 is a space
character [@] located at position 7 is a symbol
character [ ] located at position 8 is a space
character [W] located at position 9 is a consonant
character [a] located at position 10 is a vowel
character [s] located at position 11 is a consonant
character [h] located at position 12 is a consonant
character [i] located at position 13 is a vowel
character [n] located at position 14 is a consonant
character [g] located at position 15 is a consonant
character [t] located at position 16 is a consonant
character [o] located at position 17 is a vowel
```

```
        character [n] located at position 18 is a consonant
        character [ ] located at position 19 is a space
        character [^] located at position 20 is a symbol
        character [ ] located at position 21 is a space
        character [R] located at position 22 is a consonant
        character [o] located at position 23 is a vowel
        character [d] located at position 24 is a consonant
        character [r] located at position 25 is a consonant
        character [i] located at position 26 is a vowel
        character [g] located at position 27 is a consonant
        character [u] located at position 28 is a vowel
        character [e] located at position 29 is a vowel
        character [z] located at position 30 is a consonant
        character [ ] located at position 31 is a space
```

4 - Create a method called pyramid (Worth 1 point).
    Using a for loop, display to the screen, the string containing your name so that each
    loop will NOT contain the first and the last character from the previous line,
    with the length of the string being printed, and the string surrounded by square brackes []

    NOTE: Every method dealing with your name, MUST be called from main, PASSING your name to it.

    i.e:

```
 32 [George @ Washington ^ Rodriguez ]
  30 [eorge @ Washington ^ Rodriguez]
   28 [orge @ Washington ^ Rodrigue]
    26 [rge @ Washington ^ Rodrigu]
     24 [ge @ Washington ^ Rodrig]
      22 [e @ Washington ^ Rodri]
       20 [ @ Washington ^ Rodr]
        18 [@ Washington ^ Rod]
         16 [ Washington ^ Ro]
          14 [Washington ^ R]
           12 [ashington ^ ]
            10 [shington ^]
              8 [hington ]
               6 [ington]
                4 [ngto]
                 2 [gt]
                  0 []
```

5 - In a method called parsing do the following (worth 1 point):

    NOTE: Every method dealing with your name, MUST be called from main, PASSING your name to it.

    - Print your name in upper case letters.
    - Print your name in lower case letters.
    - Print your name taking all spaces out.
    - Print your name with all vowels in upper case, and all consonants in lower case.
    - Print your name backwards.
    - Print your name in ASCII values.

6 - [worth 1 point] Using a while(true) loop,
    and using the upper case alphabeth from Z to A,
    print the lower case alphabeth and its corresponding ascii values PLUS 1.
    You must terminate/exit/break this loop once you process the last letter (A).
    Note: The while(true) loop is called an endless loop because the true inside the (),
    means that the condition is always true, and it is NOT a terminating variable.

7 - [worth 1 point]
  - From main pass these 3 numbers (98, 234,   6) to a method that will print these numbers,
    then sorted them using the swap method discussed in class to make them in ascending order,
    finally print these sorted numbers.
    e.i. if you pass   (98, 234,   6)
        it will print   98, 234,   6
                         6, 98, 234

Turn in the source code on paper, and email me the source code.
Make sure the program is properly documented and aligned uniformally, looking professionally.