Program COP2250pgm3d, covering Multidimensional and Parallel Arrays,  
swap, endless while loop, %,

WORTH 8 POINTS

1 - Worth 1 point
   Passing 3 numbers call a method that will print these numbers in ascending order, 
   e.g. if you pass (98, 234, 6)
   print 6 98 234
   You must make a method called "swap", do NOT use any built-in sort

2 - Worth 1 points
   Using a while( true ) loop, print the upper case alphabeth and its corresponding ascii values for 
   each letter from Z to A
   Note: You must terminate/exit/break this loop once you process the last letter (A)

3 - Worth 2 points
   - Create a two dimensional array of 10 rows by 10 columns
   - Load each index with the multiplication of its x and y location
   - Add all the values in columns 3, 5, and 7, and print the total
   - Add all the values in rows 2, 4, and 6, and print the total
   - Subtract the total values of ( rows - columns ), and print the difference.

4 - Worth 2 points
   Implement division by 0, with error trapping, using if and while() commands, 
   make sure to use "casting" e.i. float result = (float)int/int;
   How:
   - Using a while loop, read 2 numbers from the user.
   - Using the if statement, test that the second number in not zero, if it is inform 
     the user of the error, and ask for a correct second number.
   - If the second number is NOT a zero, do the division, display all numbers and the 
     computation using labels, the result MUST have 2 decimal places,
     e.g. "The first number 10 divided by the second number 5 is 2.00"

   To exit the while loop the user must enter the value 999 for the first or the second number.

5 - Worth 1 point
   Using the loop of your choice display all numbers from 0 to 100 where "mod 5 = 3". Hint: %

6 - Worth 1 points
   Having the following TWO, ONE dimension arrays:
   one[0] = "This Java";   two[0] = "class";
   one[1] = "at ";        two[1] = "FIU";
   one[2] = "is ";       two[2] = "challenging && enjoyable";

   print the results in a parallel array format made with these two one dim arrays