1. In this exercise, you will implement several classes to represent all natural numbers consisting of at most 3 digits (all numbers from 0 to 999). You will do this by “filling in” the data and method definitions.

```
// Class representing 1-digit natural numbers (from 0 to 9)
public class Number {
  // Block A:
  // Declare a private instance variable:
  // n, 1-digit integer initialized to 0

        public Number(){
          this.n =0;
        }

        public Number(int n){
          //Block B:
          // Check that n is a 1-digit number,
          // if not, terminate using "System.exit(0)"
          // initialize instance variable to n
        }

        public Number (Number n){
         //Block C:
         // Initialize instance variable
        }

        public int getNumber(){
         //Block D:
         // Return the 1-digit number
        }

        public boolean equals(Number n){
         //Block E
         // Return true if this number is equal to n
         }

        public boolean compare(Number n){
          //Block F
          //return true if this number is greater than n
        }
        public String  toString(){
         // Block G
         // return a string consisting of this number
        }

// Class representing 2-digit numbers (from 00 to 99)
public class TwoDigitNumber extends Number {
         //Block H:
         // Declare two private instance variables
         // n1, Number representing first digit
         // n2, Number representing second digit

        public TwoDigitNumber(){
         //Block I
         // Initialize all digits to 0
         }

         public TwoDigitNumber(Number n1, Number n2){
         //Block J
         // Initialize all digits to n1 and n2
         }

         public boolean equals(TwoDigitNumber n){
         //Block K
         // Return true if this TwoDigitNumber number
         // is equal to n
         }

         public boolean compare(TwoDigitNumber n){
          //Block L
          //return true if this number is greater than n
          // Example: 21 > 11 and 10 > 08
          }

          public String toString(){
          //Block M
          // return a string consisting of this
          // TwoDigitNumber, for example 11 or 04.
          }
```

```
// Class representing 3-digit numbers (from 000 to 999)
public class ThreeDigitNumber extends TwoDigitNumber{
         //Block N:
         // Declare two private instance variables
         // n1, Number representing first digit
         // n2, TwoDigitNumber representing next 2-digits

         public ThreeDigitNumber(Number n1, TwoDigitNumber n2){
         //Block O:
         // Initialize all digits to n1 and n2
         }

         public boolean equals(ThreeDigitNumbers n){
         //Block P:
         // Return true if this ThreeDigitNumber number
         // is equal to n
         }

         public boolean compare(ThreeDigitNumber n){
         //Block Q
         //return true if this number is greater than n
         }

          public String toString(){
          //Block R
          // return a string consisting of this
          // ThreeDigitNumber, for example 111 or 003.
          }
}
```

2. Write a class TestNumbers that implements the following methods:

   ```
   public static Number getFirstDigit(int n);
   public static Number getSecondDigit(int n);
   ```

   that return a Number corresponding to the first and the second digit of a 2-digit natural number n, respectively. For example, if `n=75`, then `getFirstDigit(n)` returns the Number object corresponding to number 7 and `getSecondDigit(n)` returns the Number object corresponding to number 5.

(a) Implement a method with the signature

    `public static Number[] genNums()`

    that returns an array of 100 random numbers of either 1-digit or 2-digit, namely from 0 to 99. You don't need to generate 3-digit numbers.

(b) Implement a method with the signature

    `public static String printMax(ThreeDigitNumbers[] nums)`

    that returns the string representation of the maximum number from an array of ThreeDigitNumber numbers.

3. Use the above methods to test your program in the main method.