AP-Style FR 4.1 – Code Lab

Consider the following incomplete declaration of a `Code` class which represents a code consisting of letters and digits. The actual code is stored internally as a `String` variable, `myCode`. Portions of the code may be hidden by changing the corresponding letter or digit to an `X`. Hidden portions may later be recovered.

```
public class Code
{
       private String myCode;
       // additional instance variables

       public Code (String code)
       {
          myCode = code;
          // possibly additional statements
       }

       public String getCode()
       {
          return myCode;
       }

       // precondition: p1 >= 0, p1 < p2, p2 <= myCode.length()
       // Replace the characters in the code starting at
       // position p1 until position p2-1 inclusive with X

       public void hide (int p1, int p2)
       {
          // to be implemented
       }

       // precondition: p1 >= 0, p1 < p2, p2 <= myCode.length()
       // Restores to their original values the characters in the code starting at
       // position p1 until position p2-1 inclusive

       public void recover (int p1, int p2)
       {
          // to be implemented
       }
}
```

The methods `hide` and `recover` work as described in the comments. Note that if `hide` is called for a portion of code that is already hidden, it has no effect and if `recover` is called for a portion of the code that is already "clear", it has no effect.

Suppose the following code is created:

```
Code code = new Code ("ABCdef123ghi456jklMNO");
```

The following sequence of method calls results in the instance variable `myCode` having the indicated values.

	Value of `myCode`
`code.hide(2,7);`	`ABXXXXX234ghi456jklMNO`
`code.recover(5,9);`	`ABXXXf1234ghi456jklMNO`
`code.hide(3,14);`	`ABXXXXXXXXXXXX56jklMNO`
`code.hide(6,10);`	`ABXXXXXXXXXXXX56jklMNO`
`code.recover(5,6);`	`ABXXXfXXXXXXXX56jklMNO`
`code.recover(0,14);`	`ABCdef1234ghi456jklMNO`