

Clase Auxiliar 2

CC10A – Sección 4

30-03-2005

Profesor: Erich Reimberg
Auxiliares: Valeria Herskovic
Juan Ignacio Saba
Cristian Junge

Problema 1

Conceptos

Dados 2 números naturales de la misma cantidad de dígitos y cada uno de ellos sin dígitos repetidos, se definen los siguientes conceptos.

Fama:

Un dígito se denomina una **Fama** si se encuentra en ambos números y en la misma posición. Por ejemplo:

En 2057 y 4065, el "0" es una Fama.

En 1234 y 4321, no hay Famas.

En 1235 y 1236, hay 3 Famas; el "1", el "2" y el "3".

Toque:

Un dígito se denomina un **Toque** si se encuentra en ambos números pero en posiciones distintas. Por ejemplo:

En 2057 y 4065, el "0" es una Fama, pero el "5" es un Toque.

En 1234 y 4321, cada uno de los dígitos es un Toque (4 toques).

En 1235 y 1236, no hay Toques.

Problema

a) Escribir un método que diga cuantas famas hay entre 2 números.

```
public static int famas(int n1, int n2) { ... }
```

Escribir un método que diga cuantos toques hay entre 2 números.

```
public static int toques(int n1, int n2) { ... }
```

Nota:

Los números no pueden tener dígitos repetidos (2032 no es un número válido) y no pueden comenzar con 0. Además, si los números entregados como parámetro no tienen igual cantidad de dígitos, el método debe retornar -1.

b) Escribir un programa en el cual el computador genera un número al azar de 4 dígitos y el usuario intenta adivinarlo. Un diálogo para el programa podría ser:

```
Bienvenido!
Ya he generado mi número secreto de 4 dígitos, intenta adivinarlo.
Numero? 2056
  Respuesta: 2 Toques, 0 Famas
Numero? 4356
  Respuesta: 0 Toques, 0 Famas
Numero? 8720
  Respuesta: 1 Toques, 2 Famas
Numero? 9820
  Respuesta: Felicidades!! adivinaste!
```

Problema 2

a) Escribir un método (función) con el siguiente encabezamiento.

```
static public String borrar(String x, String y) { ... }
```

que elimine todas las apariciones de *x* en *y* . Por ejemplo, la invocación `borrar("bra", "abracadabra")` entrega el resultado "acada".

b) Utilice el método anterior en un programa que lea líneas hasta que se ingrese el *string* "fin", y que borre todas las apariciones de los *strings* ":-)" y ":-(", y luego imprima todas las líneas leídas en pantalla.