LENGUAJES DE PROGRAMACION

PRIMERA EVALUACION II TERMINO 2009-2010

Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

I (20 ptos)

Se desea extender la clase Array para que tenga un nuevo método denominado ***valores*** que genera pares de valores en el que cada uno tiene un valor del arreglo valores y el número de ocurrencias de ese valor. Cada valor diferente del arreglo debe ser generado solamente una vez y debe ser generado en el orden que aparecen en el arreglo, Un cliente debe usar un bloque con dos parámetros para accesar los pares generado por el método. Por ejemplo el siguiente código:

 ***x = [9, 7, 9, 5, 0, 6, 5, 12, 5, 9, 11, 2, 3, 0, 0, 9, 3, 10, 3, 6]***

***x.valores {|v, cuenta| print v, " ocurre ", cuenta, " veces\n"}***

debe producir la siguiente salida:

9 ocurre 4 veces

 7 ocurre 1 veces

 5 ocurre 3 veces

 0 ocurre 3 veces

 6 ocurre 2 veces

 12 ocurre 1 veces

 11 ocurre 1 veces

 2 ocurre 1 veces

 3 ocurre 3 veces

 10 ocurre 1 veces

Su método debe retornar el número de valores generados (10 en el ejemplo dado). Su método no debe cambiar el arreglo. Escriba su solución a continuación:

II (15 ptos)

Programe una clase en Ruby llamada **Estudiante**, con las siguientes propiedades

Lea un archivo de texto con el siguiente formato:

***primerNombre, segundoNombre, primerApellido, segundoApellido***

Deberá implementarse un método llamado ***comprobar*** que, una vez que se haya inicializado una determinada instancia de la clase (leído una línea de texto), identifique si todos los nombres (y apellidos) son válidos

Para que un nombre/apellido sea válido debe considerarse lo siguiente:

1. El primer carácter es una letra mayúscula
2. Los siguientes caracteres son letras minúsculas
3. No hay ningún otro tipo de carácter en la cadena

III (15 ptos)

Defina la máquina de estados, la gramática regular y la expresión regular para detectar cadenas binarias que contienen un número par de ceros.

IV (10 ptos)

**Considerando la siguiente gramática**

<S> → a <S> c <B> | <A> | b

<A> → c <A> |c

<B> → d | <A>

¿Cuál o cuáles de las siguientes cadenas pertenece al lenguaje generado por esta gramática? Demuéstrelo usando árboles de derivación ( *parse trees*)

1. abcd

2. acccbd

3. acccbcc

4. acd

5. accc

V (40 ptos)

a) Explique las Principales Características de la Arquitectura Von Neuman

b) Indique las diferencias entre las gramáticas de tipo 2 y tipo 3. Explique la máquina que representa a cada una de ellas.

c) Indique cuáles son las uniones que pueden darse en tiempo de Ejecución cuando se corre un programa en Ruby

d) Explique en qué consiste el Análisis Léxico y qué entrega como resultado

e) Explique las características de la programación imperativa y de la programación funcional.

f) Indique 5 diferencias **importantes** entre Java y Ruby

g) Grafique la estructura del Descriptor de un arreglo unidimensional, explique con un ejemplo

h) Indique la representación del número -4.75 en punto flotante según el estándar IEEE 754