

Nivelación Programación de Sistemas II

Para las personas que adeudan una unidad de la 1-3 se tendrá que realizar alguno de los dos programas listados a continuación. Para quien adeude la unidad 4, se tendrá que realizar de nuevo el proyecto final. Para quien adeude dos unidades tendrá que realizar forzosamente dos programas. Quién tiene más de dos unidades reprobadas no tiene derecho a nivelación y repite curso o examen especial (dependiendo del caso).

Los programas deberán ser entregados antes del viernes 8 de junio de 2007, para tener derecho a extraordinario sólo se debe adeudar una unidad. Si se llega el caso de extraordinario deberá entregarse los mismos problemas antes del miércoles 13 de junio.

Problema 1

Dado un archivo en HTML el programa deberá determinar y obtener todos los enlaces. Dichos enlaces están identificados por la etiqueta <a> el cual tiene la siguiente sintaxis:

```
<a href="URL" OTROS_PARAMETROS>TEXTO</a>
```

El sistema deberá identificar todos los URLs y compararlas con una lista, en caso de que la URL no exista en la lista deberá difuminarla. Difuminar un enlace consiste en quitar el vínculo de un enlace; es decir, dejar solo el TEXTO. La liga anterior difuminada quedaría:

TEXTO.

El sistema podrá ser implantado en cualquier lenguaje, de preferencia Java con XML o bien FLEX/LEX.

Ejemplo: sea el siguiente código en HTML

```
<html>
....
<a href="http://www.itmorelia.edu.mx">Tec Morelia</a> ...
<a href="mailto:a@i.com" target="_BLANK">Buzón</a> ...
<a href="ftp://pub.itm.mx/xml.zip">XML</a> ...
<a href="ftp://pub.itm.mx/lyx.tar.gz">LYX</a> ...
.....
</html>
```

Si el archivo de direcciones es el siguiente:

<http://www.itmexicali.edu.mx>
<http://www.itmorelia.edu.mx>
<http://www.teczamora.edu.mx>
<http://www.mimorelia.com>

<ftp://pub.microsoft.com>
<ftp://pub.itm.mx/lyx.tar.gz>

El archivo con las ligas difuminadas quedaría así:

```
<html>
....
<a href="http://www.itmorelia.edu.mx">Tec Morelia</a> ...
Buzón
XML
<a href="ftp://pub.itm.mx/lyx.tar.gz">LYX</a> ...
.....
</html>
```

Problema 2

Crear un programa para verificar que los métodos optimizados son mejores que los métodos normales. A continuación se muestran los métodos normales y optimizados:

Método 1

```
for(int i=0; i<1000000;i++)
    int b = 3*b;
```

Optimizado:

```
int b;
for(int i=0; i<1000000;i++)
    b = 3*b;
```

Método 2

```
for(int i=0; i<size(); i++)
    a = (b+c) / i;
```

Optimizado:

```
int tmp = b+c;
int s = size();
for(int i=0; i<s; i++)
    a = tmp / i;
```

Método 3

```
b = Math.abs(a) * c;
d = e / (Math.abs(a) + b);
```

Optimizado:

```
int tmp = Math.abs(a);
b = tmp * c;
d = e / (tmp + b);
```

Método 4

```
for (int i=0; i <1000; i++)  
    a = obj.b * i;
```

Optimizado:

```
int localb = obj.b;  
for (int i=0; i <1000; i++)  
    a = localb * i;
```

Método 5

```
for(int i=0; i <1000, i++)  
    a[i] = 25;
```

Optimizado:

```
for(int i=0; i <100; i++) {  
    a[i++] = 25;  
    a[i++] = 25; //8 veces más  
}
```

Los métodos se deberán codificar en un lenguaje para dispositivos móviles (J2ME/ .NET Compact Framework), los métodos se deben repetir en un ciclo (el usuario debe decidir el número de iteraciones) y se debe calcular un promedio de cada método normal y optimizado, sacando el porcentaje de diferencia. Dentro de cada método se debe tomar el tiempo de inicio y el final que duró el proceso.

Ejemplo de corrida:

Método	Normal	Optimizado	Diferencia	Porcentaje
1	100	125	25	25%
2	100	101	1	1%
3	...			
4				
5				

Aparte del dispositivo móvil se deberá realizar el programa en un lenguaje para PC normal, o bien probarlo con otro lenguaje de programación para dispositivo móvil, o bien una comprobación del programa con distintos dispositivos móviles y emuladores.