

Nombre:	
C.I.:	

# TECNÓLOGO EN INFORMÁTICA

## ESTRUCTURAS DE DATOS Y ALGORITMOS

### Segundo Parcial

28 de Noviembre de 2008

#### Ejercicio 1 (14 puntos)

a) (2 pts.) Describa la estructura para el TAD archivos que utilizo para resolver el segundo laboratorio de la materia (edafs). (Se pide que de los campos de `struct` `NodoArchivos`, y que comente cada campo indicando para que es utilizado)

b) (3pts.) Escriba una especificación mínima para el TAD lista de cadena de caracteres( `LCadena`), dando un conjunto mínimo de operaciones constructoras, selectoras, predicado y destructoras.

Nota: asuma que el tipo cadena esta definido de la siguiente forma

```
typedef char* Cadena
```

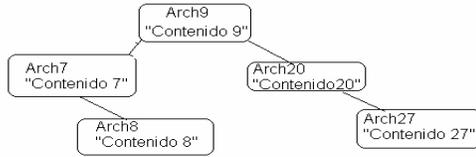
c) (9pts.) Accediendo directamente a la representación del TAD Archivos, extienda este TAD agregando una nueva función auxiliar

```
void CambiarContenidos(Archivos as, LCadena& nomarchs, LCadena& contenidos)
```

Que recibe dos listas ordenadas alfabéticamente en orden decreciente, en donde cada nodo de la lista “nomarchs” contiene un nombre de archivo en cada posición, y la lista “contenidos” en cada posición contiene un nuevo contenido para el archivo cuyo nombre se encuentran presente en la lista “nomarchs” en la misma posición. Es decir, en la posición *i* de la lista “contenidos” se encuentra el nuevo contenido para el nombre de archivo que se encuentra en la posición *i* de la lista “nomarchs”.

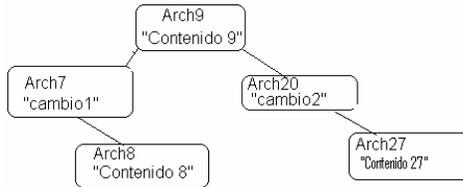
Caso	nomarchs	contenidos
1	[“Arch7”, “Arch20”]	[“cambio1”, “cambio2”]
2	[“Arch27”]	[“cambio27”]

**Ejemplos :** sea la siguiente instancia de archivos como muestra la figura 1.

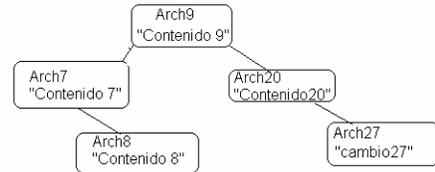


**Figura 1.**

**Resultado Caso 1:**



**Resultado Caso 2:**



**Notas:**

- No se pueden utilizar funciones ni TADS auxiliares, solamente puede utilizar las funciones del TAD LCadena que especifico en la parte b y el TAD Archivo.
- No se pueden utilizar variables auxiliares de tipo LCadena en el procedimiento.
- No se puede recorrer más de una vez el contenido de las listas.
- Para acceder a los archivos contenidos en la estructura puede utilizar las siguientes operaciones del TAD Archivo:

```

void setContenidoArchivo (Archivo & a, char * contenido);
char * getNombreArchivo (Archivo a);
  
```

**Ejercicio 2 (13 puntos)**

- a) (2 pts) De una especificación mínima para el TAD Árbol binario de búsqueda de naturales (ABBNat). Dando un conjunto mínimo de operaciones constructoras, selectoras, predicado y destructoras.
- b) (5pts) De un representación e implemente todas las operaciones para el TAD árbol binario de búsqueda de la parte a).
- c) (6 pts) Sin acceder a la representación de ABBNat, implemente la función auxiliar

```

int SumaNivel(ABBNat a,int niveln);
  
```

que devuelve la suma de todos los naturales que se encuentran en el nivel "niveln" del árbol a.

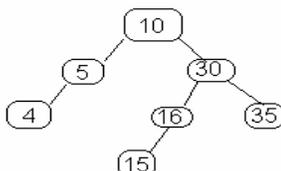
**Nota:**

- No se puede utilizar ninguna función auxiliar.

**Ejemplos:**

```

SumaNivel( a,1)=10
SumaNivel( a,2)=35
SumaNivel( a,4)=15
  
```



**Figura 2**

### Ejercicio 3 (20 puntos)

Para cada uno de los siguientes fragmentos de programa de los ordenes O del tiempo de ejecución de los mismos. Muestre el razonamiento aplicado para calcular los mismos.

Puede utilizar la sumatoria:

$$\sum_{i=1}^n i^2 = n(n+1)(2n+1)/6$$

y aproximar la  $\sum_{i=1}^n i^k$  con  $n^{k+1}/(k+1)$

(4 ptos) 

```
sum=0;
for(i=1;i<=n;i++)
    for (j=1,j<=n;j++)
        sum++;
```

(4 ptos) 

```
sum=0;
for(i=1;i<=n;i++)
    for (j=1;;j<=i;j++)
        sum++;
```

(6 ptos) 

```
sum=0;
for(i=1;i<=n;i++)
    for (j=1;j<=i*i;j++)
        if (j - i == 0) then
            for(k=1;k<=j;k++)
                sum++;
```

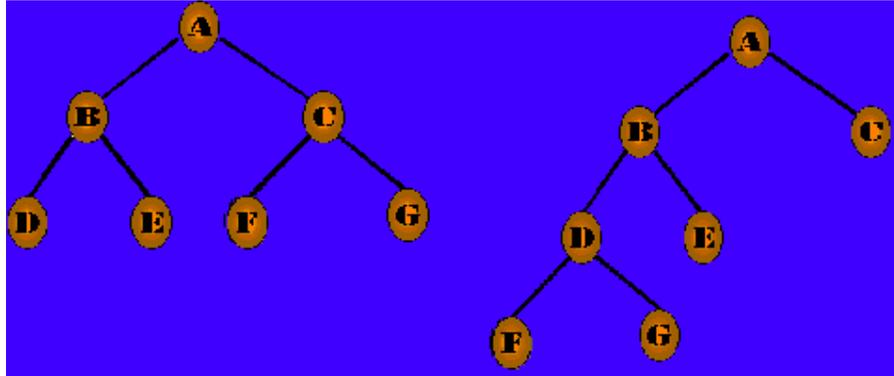
(6 ptos) 

```
sum=0;
for(i=1;i<=n;i++)
    for (j=1;;j<=i*i;j++)
        if (j % i == 0) then
            for(k=1;k<=j;k++)
                sum++;
```

### Ejercicio 4 (13 puntos)

Dados dos árboles binarios a, y b , y la siguiente definición:

ÁRBOLES BINARIOS SEMEJANTES : Dos árboles binarios son semejantes si tienen el mismo número de nodos y los valores de los nodos del primer árbol son los mismos que los valores de los nodos del segundo árbol, sin importar que no tengan las mismas relaciones entre ellos.



Implemente la función bool Semejantes(Arbol a, Arbol b), accediendo a la representación del tipo Arbol(árbol binario de caracteres) que devuelve verdadero si los dos árboles son semejantes. Para resolverlo se pueden utilizar TADs y funciones auxiliares, pero debe implementar todas las funciones que utilice. Comente los pasos más importantes de su solución