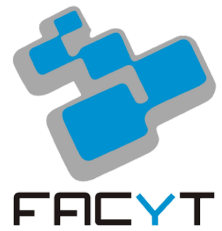




UNIVERSIDAD DE CARABOBO  
Facultad Experimental de Ciencias y Tecnología  
Departamento de Computación  
Unidad Académica de Algoritmos y Programación  
CAO403: Programación II



**Profesor: Álvaro Espinoza**

**Taller II**  
**Estructuras Jerárquicas: Árboles**  
**(Valor: 30%)**

**1. Instrucciones**

- a. El ejercicio propuesto requiere la lectura de datos de la entrada estándar (standard input) y la escritura de resultados en la salida estándar (standard output).
- b. Para el desarrollo del taller debe utilizar el lenguaje de programación C++, junto con sus librerías estándar. Para la compilación de sus códigos fuentes, debe realizarla por medio de un archivo **makefile**, el cual deberá entregar junto con sus códigos fuentes. De no poseer makefile la entrega, el taller no será revisado.
- c. Su código debe estar debidamente comentado.
- d. Debe utilizar lowerCamelCase para las variables, y UpperCamelCase para la creación de tipos de datos (en caso de necesitarlos).
- e. Se permite el uso de las bibliotecas de STL que proporciona C++.
- f. Debe utilizar sus propias librerías de árboles creadas por usted.
- g. Debe modularizar su código de tal forma que se evite el “código spaghetti”.
- h. El archivo makefile debe generar un ejecutable con el nombre “arboles”, con el fin de facilitar las labores de corrección de su taller, el no hacerlo generará puntos menos en su nota final del taller.
- i. La fecha de entrega máxima es el día viernes 15 de noviembre de 2020 a las 11:59pm. Se restará un punto por cada hora de retraso.
- j. Debe enviar un archivo comprimido .zip con los archivos necesarios y un documento .txt con su nombre y cédula al correo: aespinoza3@protonmail.com

**2. Enunciado**

Una antigua civilización tenía un emperador de nombre Viktorsés Mendozamón, el cual fue escogido por elección divina por parte de los dioses a muy temprana edad y cuyo poder sería heredado por sus futuros hijos en plan de fundar una dinastía eterna. Sin embargo, el emperador después de 10 años de mandato se aburrió de gobernar y quería entregar el trono. Sin embargo, al no tener hijos decidió sortear el mandato.

Para entregar el trono decidió que escogería a sus  $m$  soldados mejor preparados en combate ya que desea que el sucesor del trono sea un honorable guerrero tal como Viktorsés lo fué. Por otra parte, sólo se escogió a los guerreros cuya ascendencia y descendencia familiar siguiera un patrón de máximo dos hijos por individuo, ya que esto permitiría un mejor linaje con menos mezclas de sangre de distintas familias. El proceso de escogencia consistió en lo siguiente: por cada familia, cada miembro se enumerará de 1 a  $n_i$ , donde  $n_i$  es el número de miembros de la familia del guerrero  $i$ . El consejero del emperador, de nombre Alvaronus Espinozifis le asignó a cada guerrero un conjunto  $Q_i$  de números tal que  $(1 < |q| \leq n)$  y cada  $q_i$  perteneciente a  $Q_i$  es un número asociado a un miembro ya previamente enumerado de esa familia. Entonces, el emperador necesita calcular para cada familia:

$$\left( \sum_{\{u,v\}} u \cdot v \cdot dist(u, v) \right)$$

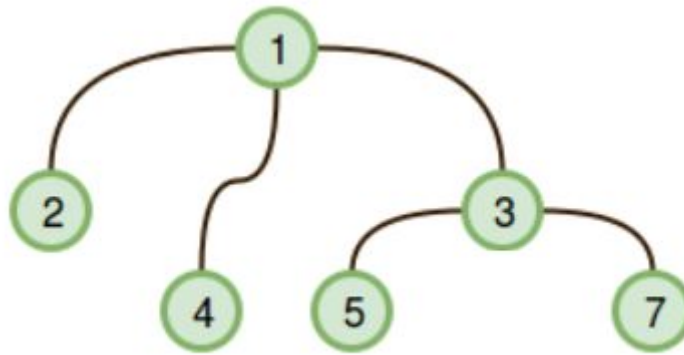
Donde:

- $\{u, v\}$  denota un par no ordenado de familiares que pertenecen al conjunto  $Q_i$ .
- $dist(u, v)$  representa el número de saltos entre un familiar  $u$  y un familiar  $v$  en el árbol familiar.

Entonces, luego de hacer este cálculo, se escoge al guerrero cuya familia haya obtenido el mayor puntaje. Para realizar el cálculo, Alvaronus te escoge a tí, que eres una persona de confianza para el emperador, para que realices los cálculos necesarios y definas quién será el nuevo emperador.

### **Explicación del cálculo por familia:**

Dado un árbol familiar:



Y el conjunto  $Q = \{4, 5, 7\}$  asociado a esa familia, hay tres pares que se pueden formar con ese conjunto:  $[4, 5]$ ,  $[4, 7]$  y  $[7, 5]$ . La distancia de 4 a 5 es 3, la de 4 a 7 es 3 y la de 5 a 7 es 2. Ahora se suma:

$$(4 \cdot 5 \cdot \text{dist}(4, 5) + 4 \cdot 7 \cdot \text{dist}(4, 7) + 5 \cdot 7 \cdot \text{dist}(5, 7))$$

$$\Rightarrow (4 \cdot 5 \cdot 3 + 4 \cdot 7 \cdot 3 + 5 \cdot 7 \cdot 2)$$

$$\Rightarrow 214$$

En este caso, 214 sería el valor calculado para esa familia, el resto es calcular las demás y escoger el mayor. (De alguna forma Alvaronus está manipulando la elección del nuevo líder).

### 3. Formato de Entrada

La primera línea será un entero  $c$  que representa el número de casos de prueba, Seguido por cada caso que contendrá una primera línea con un entero  $m_j$  que representa el número de familias del caso, y luego, por cada familia, habrá un entero  $n_i$  que representa el número de miembros de esa familia, seguido por dos líneas que contendrán los recorridos en (preorden, inorden) o (postorden, inorden), para la construcción de los árboles binarios. Cada línea comenzará con una palabra que indica el tipo de recorrido (PREORDEN, INORDEN, POSTORDEN) seguida por  $m$  caracteres que describen el recorrido correspondiente.

Luego de las dos líneas, vendrá una línea siguiente con  $k$  caracteres que representarán los elementos del conjunto  $Q$ , los cuales serán enteros que pertenecen al árbol definido en las dos líneas anteriores, y no se repiten.

**IMPORTANTE:** los árboles son en un principio de caracteres, es importante hacer un mapeo del recorrido INORDEN para enumerar cada nodo. Por ejemplo, si se tiene el recorrido inorden “d b h e a f i c g” entonces, se obtiene la secuencia “1, 2, 3, 4, 5, 6, 7, 8, 9” donde el 1 corresponde al caracter ‘d’, el 2 corresponde a ‘b’ y así sucesivamente. De no hacer este mapeo correctamente los resultados se verán afectados y por lo tanto la nota del taller.

#### 4. Ejemplo de Entrada

```
1
3
9
PREORDEN a b d e h c f i g
INORDEN d b h e a f i c g
1 3 5 7
7
POSTORDEN d b f g e c a
INORDEN d b a f e g c
4 3 2 1
4
POSTORDEN d c b a
INORDEN d c b a
1 3
```

#### 5. Formato de Salida

La salida será un formato sencillo consistente en  $c$  líneas que contendrán un número entero que representan el número de la familia ganadora por cada caso. Para el ejemplo de entrada anterior hay tres familias, se enumeran de arriba hacia abajo desde 1 a  $m_j$ .

**Nota:** queda para el estudiante crear sus propios casos de prueba.

*“La tarea del programador no es sólo escribir un programa, sino que su tarea principal es dar una prueba formal de que el programa que propone cumple la especificación funcional.”*  
*- Edsger Dijkstra*