

2° PARCIAL DE ALGORITMICA Y PROGRAMACION II - 2021

1) Agregar a la clase **ArrayList.java** el siguiente método:

```
/**
 * Retorna una lista nueva con los elementos especificados en fromIndex inclusive
 * (incluye la posición dada) y toIndex exclusive (es uno menor a la posición dada).
 * Si fromIndex y toIndex son iguales, retorna una lista vacía.
 *
 * @param fromIndex elemento inicial inclusive (incluye la posición dada) de subList
 * @param toIndex elemento final exclusive (es uno menor a la posición dada) de subList
 * @return una lista nueva con sus elementos dentro del rango especificado
 * @throws IllegalArgumentException si fromIndex > toIndex
 * @throws IndexOutOfBoundsException si (fromIndex < 0 || toIndex > size)
 *
 * Ejemplo:
 *
 * Dada la lista: [A, B, C, D, E]
 *
 * subList(2,4)
 *
 * Retorna la lista: [C, D]
 *
 * Notar que incluye el elemento que está en la posición 2 (incluye) y no incluye el
 * elemento que está en la posición 4 (exclusive)
 */
List<E> subList(int fromIndex, int toIndex);
```

Realizar programas que llamen a subList con distintos parámetros probando las distintas condiciones que se pueden presentar y atrapar sus excepciones. Probar por ejemplo si el primer argumento es menor al segundo, los dos argumentos son iguales, el primero argumento es mayor al segundo, el primer argumento es negativo, el segundo argumento es mayor al tamaño de la lista, etc.

2) Realizar un programa que contenga los siguientes métodos:

```
/**
 * Retorna una lista posicional con los elementos pasados en un array de String
 */
public static PositionalList<String> cargarPL(String i[])

/**
 * Verifica si los elementos de una lista posicional son palíndromos
 */
public static boolean palindromo(PositionalList<String> pl)
```

Invocar los métodos creados con distintos datos para probar su funcionamiento. Por ejemplo:

```
String s[] = {"N", "E", "U", "Q", "U", "E", "N"};
PositionalList<String> p = cargarPL(s);
System.out.println(palindromo(p));
```

Nota: no modificar ningún TAD.

3) Realizar un constructor para la clase **LinkedBinaryTree.java**

```
/**
 * Crea un árbol a partir de un árbol binario pasado en un array
 *
 * @param array árbol binario
 *
 * Por ejemplo dado el siguiente array
 *
 * {"/", "*", "+", "+", "4", "-", "2", "3", "1", null, null, "9", "5",
 * null, null }
 *
 * Se crea un árbol cuyo recorrido es el siguiente:
 *
 * Por nivel: / * + + 4 - 2 3 1 9 5
 * Inorder: 3 + 1 * 4 / 9 - 5 + 2
```

```

*
*/
public LinkedBinaryTree(E array[])

```

Para su desarrollo va a necesitar guardar las posiciones de los elementos que agrega en el árbol. Utilizar una lista de la longitud del array (recibido como argumento en el constructor) con todos sus elementos en null. Utilizar `set` para almacenar las posiciones y `get` para obtener los mismos.

Por ejemplo:

```

List<Position<E>> p = new ArrayList<Position<E>>(array.length);
for (int i = 0; i < array.length; i++)
    p.add(i, null);

```

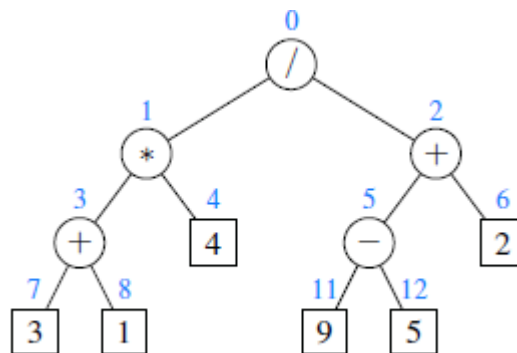
Dado un arreglo que representa un árbol binario es fácil encontrar el padre aplicando la siguiente fórmula:

(Posición del elemento - 1) división entera 2

Por ejemplo, dado el siguiente arreglo:

/	*	+	+	4	-	2	3	1	NULL	NULL	9	5	NULL	NULL
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Que corresponde al siguiente árbol:



La posición del padre del elemento que está en la posición 5 (elemento -) es:

$$(5 - 1) / 2 = 2 \text{ (elemento +)}$$

La posición del padre del elemento que está en la posición 12 (elemento 5) es:

$$(12 - 1) / 2 = 5 \text{ (elemento -)}$$

TEORIA

1. ¿Qué es la interfaz de un Tipo Abstracto de Datos (TAD)?
2. Explique cuál es la diferencia entre recursión lineal, binaria y de cola.
3. Defina y explique el funcionamiento del TAD Lista Posicional.
4. Indique la definición formal de un árbol de forma recursiva.

Nota: se tendrá en cuenta la redacción correcta de las respuestas, que se entiendan los conceptos y que sintácticamente estén bien construidas.

IMPORTANTE:

1. Los enunciados no deberían dejar lugar a dudas de los ejercicios a resolver y preguntas a responder. De todas maneras si surge alguna consulta del enunciado enviar la misma al **Foro** que está en la **sección Evaluación**. Las preguntas serán respondidas dentro de los 30 minutos de realizadas.
2. Enviar un WhatsApp al grupo **solo** si tienen algún inconveniente (no pueden entrar al foro, no reciben la respuesta en el tiempo indicado, no pueden subir las soluciones, etc.)
3. Para la parte práctica del parcial, subir solamente los archivos con extensión .java.
4. Para la parte teórica del parcial, subir un solo archivo de Word. (con extensión doc o .docx)