TP - Unidad 02
Algoritmos de Búsqueda en Texto- Parte B

Ejercicio 1

1.1) Crear en un proyecto la clase ExactSearch con el método de clase:

```
public class ExactSearch {
     public static int indexOf(char[] query, char[] target)

     {
     …
     }
}
```

El mismo recibe 2 arreglos de chars (no strings) y devuelve el índice de la primera aparición de query en target. Si no hay tal aparición, devuelve -1.
No utilizar ningún método de Java (es solo buscar en un arreglo!)

Casos de uso (realizar Testeos de Unidad para estos y otros casos)

```
char[] target= "abracadabra".toCharArray();
char[] query= "ra".toCharArray();
System.out.println( ExactSearch.indexOf( query, target) );  //2

char[] target= "abracadabra".toCharArray();
char[] query= “abra".toCharArray();
System.out.println(ExactSearch.indexOf( query, target) );  //0

char[] target= "abracadabra".toCharArray();
char[] query= “aba".toCharArray();
System.out.println(ExactSearch.indexOf( query, target) );  //-1

char[] target= "ab".toCharArray();
char[] query= “aba".toCharArray();
System.out.println(ExactSearch.indexOf( query, target) );  //-1

char[] target= “xa".toCharArray();
char[] query= “aba".toCharArray();
```

System.*out.println(ExactSearch.indexOf(query, target)); //-1*

char[] target= "abracadabras".toCharArray();
char[] query= “abras".toCharArray();
System.*out.println(ExactSearch.indexOf(query, target)); //7*

1.2) ¿Qué complejidad espacial tiene el algoritmo propuesto?

1.3) ¿Qué complejidad temporal tiene el algoritmo propuesto?

1.4) Este comportamiento es el que ofrece el método indexOf de Java en la clase String. Buscar como está implementado en Java el código de String.indexOf. Compararlo con el que Uds. propusieron en el ítem 1.1

Ejercicio 2

Escribir la clase KMP para implementar el método de búsqueda exacta de Knuth-Morris-Pratt.

2.1) Escribir el método de clase nextComputation(char[] query) que calcule la tabla de Next para un query.

2.2) Qué complejidad espacial tiene dicho cálculo?

2.3) Qué complejidad temporal tiene dicho cálculo?

Ejercicio 3

En la clase KMP:

3.1) Escribir el método de clase `indexOf(char[] query, char[] target)` que implemente el algoritmo de Knuth-Morris-Pratt y usa la tabla de Next.

3.2) Qué complejidad espacial tiene dicho cálculo?

3.3) Qué complejidad temporal tiene dicho cálculo?

Ejercicio 4

4.1) Agregar a la clase KMP el método

```
static   ArrayList<Integer> findAll(char[] query, char[]target)
```

que devuelve todas las posiciones en que ocurre query dentro de target. Nunca el puntero al target debe volver hacia atrás!!!
Aprovechar next[] lo máximo que se pueda si se logra más de una coincidencia solapada también.

Casos de Uso:

```
char[] query= "no".toCharArray();
char[] target= "sino se los digo no se si es nocivo".toCharArray();
```

KMP.findAll(query, target) debería devolver una instancia de arraylist con las posiciones 2, 17, 29

```
char[] query= "ni".toCharArray();
char[] target= "sino se los digo no se si es nocivo".toCharArray();
```

KMP.findAll(query, target) debería devolver un empty arraylist

4.2) Escribir los testeos de unidad correspondientes.

4.3) Resiste el caso de solapamiento? Caso contrario, arreglarlo.

```
char[] query= "aaa".toCharArray();
char[] target= "aaabaaaaab".toCharArray();
```

KMP.findAll(query, target) debería devolver una instancia de arraylist con las posiciones 0, 4, 5, 6

Ejercicio 5

Una empresa de marketing ha decidido colocar un cartel (irregular) que abarca varios edificios consecutivos en la misma mano de una avenida principal de una importante ciudad.

Se tiene la lista de edificios en la avenida principal. De cada edificio se conoce la altura que ofrece para colocar carteles.

Escribir una aplicación que calcule de la forma más eficiente posible si es posible distribuir el cartel entre edificios consecutivos y devuelva esas posiciones. Para que un cartel pueda ser colocado en un edificio su altura tiene que coincidir con la altura que el edificio provee para colocar carteles.

Ej: la avenida principal tiene los siguientes edificios y sus alturas ofrecidas:

10	5	3	5	3	5	3	5	5	5	3	5	3	5

Si el cartel se divide en 4 partes que requiere las siguientes alturas: 5, 3, 5 y 3

Las posibilidades de colocarlo en 1, 3 y 9

Ej: la avenida principal tiene los siguientes edificios y sus alturas ofrecidas:

10	5	3	5	3	5	3	5	5	5	3	5	3	5

Si el cartel se divide en 3 partes que requiere las siguientes alturas: 5, 5 y 5

Las posibilidades de colocarlo en 7

Ej: la avenida principal tiene los siguientes edificios y sus alturas ofrecidas:

10	5	3	5	3	5	3	5	5	5	3	5	3	5

Si el cartel se divide en 3 partes que requiere las siguientes alturas: 5, 2 y 5

No existe posibilidad.