Trabajo Práctico II

1. Motivación del problema

Muchas aplicaciones, incluyendo editores de texto, procesadores, clientes de e-mail incluyen un corrector ortográfico. El trabajo de un corrector ortográfico es relativamente simple: dada una lista de palabras correctas (la cual llamaremos nuestro *Universo*) y un texto de entrada; el corrector ortográfico debe avisar todos los errores de ortografía, es decir, todas las palabras no encontradas en el texto.

Cuando se encuentra un error de ortografía, un buen corrector debería sugerir palabras en nuestro *Universo* que sean correctas y, se *aproximen* a la palabra erronea.

Veamos un ejemplo, supongamos que en nuestro *Universo* están las palabras:

- vaso,
- caso,
- bazo,
- brazo,
- craso,

y recibimos una entrada que contiene la palabra `vazo` entonces, dado que es un error de ortografía nos tendría que sugerir que, tal vez, la palabra que se intentó ingresar fue: vaso o caso o bazo.

Por supuesto que la tarea de un Corrector Ortográfico se centra en buscar palabras en un *Universo* grande. Una búsqueda eficiente es crítica para la performance del corrector ortográfico, dado que buscar en el *Universo* no es sólo buscar las palabras de la entrada sino posiblemente, cientos de sugerencias para cada error ortográfico que se encuentre.

2. Objetivos

El programa que se tendrá que realizar, será un corrector ortográfico rudimentario: éste leerá un *Universo* de un archivo (provisto por la Cátedra), tomará un archivo de entrada y, por cada error ortográfico encontrado, el programa debera entregar una lista de sugerencias (si es que hay alguna).

3. Detalles de la implementación

Existen dos correctores ortográficos populares en Unix/Linux. Uno se llama **ispell** (`https://www.cs.hmc.edu/~geoff/ispell.html`), el otro es un un programa llamado **aspell** (`http://aspell.net/`) que tiene licencia GNU. Ambos usan técnicas similares para sugerir palabras. Mientras están chequeando la ortografía de un archivo de entrada, si una palabra no esta presente en el *Universo*, usan 5 técnicas para generar posibles sugerencias. Cada posible sugerencia es buscada en el *Universo*; si es encontrada se agrega como sugerencia. Las cinco técnicas usadas son:

- Intercambiar cada par de caracteres adyacentes en la palabra

- Entre cada par de caracteres de la palabra, insertar cada letra de la 'A' a l a 'Z'
- Eliminar cada caracter de la palabra
- Reemplazar cada caracter de la palabra con cada letra de la 'A' a l a 'Z'
- Separar la palabra en un par de palabras agregando un espacio entre cada par de caracteres adyacentes en la palabra. En este caso, se debería sugerir si ambas palabras pertecen al *Universo*.

El programa a entregar deberia generar sugerencias usando estas cinco técnicas. El método main debería permitir pasar el nombre del archivo de entrada, el nombre del archivo de salida. Es decir, sería algo asi:

main archivoEntrada archivoSalida

3.1. Formato de archivo de entrada

El archivo de entrada puede tener múltiples líneas de texto. Cada una de ellas pueden tener varias palabras, las cuales estarán separadas por un solo espacio; también pueden estar los siguientes carácteres especiales inmediatamente a continuación de una palabra: ":", ";", ",", ".", ¿", ¡". También, en el archivo de entrada podrían existir líneas vacías. Considere que las palabras podrían estar escritas con mayúsculas, minúsculas, o una mezcla de ambas.

3.2. Formato de archivo de salida

Por cada palabra no incluída en nuestro *Universo* que encontremos en el archivo de entrada, se deberá imprimir un bloque como se indica a continuación. Al final de cada bloque se deberá imprimir una línea vacía para separar.

Si no hay sugerencias para realizar, el bloque que se imprima deberá decir:

```
Linea N, "PALABRA"no esta en el diccionario
```

Si tenemos sugerencias para realizar, el bloque debe decir:

```
Linea N, "PALABRA"no esta en el diccionario.
Quizas quiso decir: SUGERENCIA1, SUGERENCIA2, ..., SUGERENCIAN
```

3.3. Ejemplo de entrada y salida

Entrada: `Brazo largo; Vasoh`
Salida: `Linea 1, "largo"no esta en el diccionario`

```
Linea 1, "Vasoh"no esta en el diccionario
Quizas quiso decir: vaso
```

4. Evaluación

Para la evaluación del Trabajo Práctico se tomarán en cuenta los siguientes elementos:

a) Estructura de datos usada

b) Algoritmo propuesto

c) Eficiencia

d) Calidad del código entregado