



A la població d'Iltirta s'ha trobat un llistat de les paraules que normalment s'utilitzen en les inscripcions, i s'ha distribuït per totes les poblacions on s'envien missatges. Això ha fet que no calgui afegir informació redundant als texts per si pateixen modificacions, ja que els canvis que es poden produir són pocs.

El text d'entrada és un text escrit en un dialecte de l'Íber (alguns en diuen runes), i el que hem d'aconseguir és corregir, els petits canvis que hagin patit, amb l'ajuda d'un diccionari. Els canvis que una paraula pot patir són 3: afegir un nou caràcter, eliminar un caràcter o modificar un caràcter.

Per exemple, donat el text *“eri frændo.”*, i un diccionari amb les paraules *“einhendr er erfidi fé feigra ferd flagella flædar frænda fylkir”*, el text corregit és *“er frænda.”*, ja que són les paraules més properes del diccionari, en la primera paraula hem d'eliminar el darrer caràcter, i en la segona paraula hem de canviar el darrer caràcter.

## Requeriments

- Heu d'implementar un programa que corregeixi les paraules d'un text, donat un diccionari.
- El text i el diccionari estan en format UTF-8, per permetre caràcters que no són ASCII.
- El programa principal ha de ser en `python`, al que se li passa per paràmetre el nom del fitxer. El vostre programa ha de corregir les paraules, és a dir, trobar la paraula de distància mínima al diccionari.
- Heu de fer un informe que contingui, sobre l'algorisme més important: pseudo-codi, cost teòric i experimental. Tingueu en compte com varien els costos segons la longitud del text.

## Arguments i parametres

L'execució de l'aplicació haurà de seguir la següent sintaxi:

```
$ ./corrector.py diccionari.txt text-amb-errors.txt text-corregit.txt nombre-editions
```

Els arguments del programa són els següents:

**fitxer diccionari.txt** en format UTF-8, que conté un llistat de paraules ordenades separades per espais.

**fitxer text text-amb-errors** : Fitxer d'entrada amb el text que pot contenir possibles errors, en format UTF-8.

**fitxer text-corregit** : Fitxer de sortida amb les paraules corregides, en format UTF-8.

**nombre-edicions.txt** : Nombre mínim d'edicions necessàries per recuperar el text original.

Si no hi ha arguments opcionals, l'aplicació llegeix de l'entrada estàndard i escriu a la sortida estàndard.

Si en el fitxer `text.txt` posem el text de l'exemple anterior:

```
$ cat original
eri frændo.
$ ./corrector.py diccionari.txt original.txt corregit.txt nombre.txt
$ cat corregit.txt
er frænda.
```

## Entorn

Se us proporcionen els següents programes i fitxers:

**Makefile** Podeu testejar fent `make test` i teniu un esquema per la part experimental `make experiment`.

**textgenerator.py** Genera un text amb les paraules del diccionari amb el nombre de paraules que li digueu.

**noise.py** Fa el nombre màxim de modificacions (afegir caràcter, eliminar caràcter o canviar caràcter) per paraula que li digueu al fitxer que li passeu.

## Avaluació

En l'informe de la pràctica, els algorismes principals, les taules i les gràfiques han d'estar comentats.

Els programes han de compilar sense errors ni warnings, i si el text ha estat xifrat utilitzant una clau entre les longituds donades, i la paraula coneguda apareix en el text en clar, ha de trobar sempre una clau (es possible si feu proves, que tingueu falsos positius, no passa res).

La baremació de la pràctica (sobre 10) serà la següent:

**Costos** 3 punts, anàlisi de costos teòrics (recursiu i iteratiu) i empírics dels algorismes;

**Disseny** 3 punts, recursiu i iteratiu (es tindrà en compte el cost de l'algorisme);

**Implementació** 4 punts

- 1 punt, iteratiu en C
- 1 punt, recursiu en python
- 1 punt, bones pràctiques de programació
- 1 punt, utilització dels recursos del llenguatge de programació.
- 1 punt extra, recursiu en haskell.

Només per als *alumnes de pla antic*, s'haurà de fer un estudi de la viabilitat de fer l'algorisme principal amb programació dinàmica.

## Enviament

L'assignació és per parelles, i representa un 20% de la nota final. Presenteu la pràctica al Campus Virtual de la UdL amb dos fitxers: un pdf per a l'informe, i un arxiu comprimit, `tgz` o `zip` (no s'admeten altres formats de compressió), per al codi font.

Els programes que s'hagin de compilar ho han de fer amb:

```
$ make
```

Els programes es testejen fent:

```
$ make test
```

No s'avaluarà cap pràctica que no tingui tots els fitxers per a poder fer la compilació amb `make` i les proves de test amb `make test`.

La pràctica podria ser verificada individualment el primer dia de laboratori després de l'entrega.