

Gestor de textos i cites

Enunciat de la Pràctica de PRO2
Primavera 2016

20 d'abril de 2016

1 Introducció

Volem desenvolupar una aplicació per a gestionar informació de textos i cites associades a aquest textos. Els textos poden correspondre a llibres, articles, manuals, etc., però aquí entendrem un text com un contingut que té un autor i un títol. El contingut és una seqüència de frases. **Una frase és una seqüència de paraules separades opcionalment per comes (,) punts i comes (;) o dos punts (:), i acabades en punt (.), interrogant de tancar (?) o admiració de tancar (!).** Entenem una cita com una o més frases consecutives d'un text. Cada frase té un número de frase associat que indica la seva posició en el text: la primera frase es la número 1, la segona frase és la número 2, i així successivament. El número de frase és una forma més robusta de referir-se al contingut d'un text respecte el número de pàgina, que està fortament determinat pel format de l'edició. El sistema proveeix una sèrie d'operacions per fer consultes sobre el contingut dels textos; per exemple, mostrar les frases que contenen un determinat conjunt de paraules, a l'estil dels cercadors d'internet i fins i tot més elaborades. Algunes d'aquestes operacions serveixen per definir cites.

Els sistema interactua amb l'usuari mitjançant una línia de comandes basada en el canal d'entrada i sortida estàndard. Les comandes de consulta es distingeixen de la resta perquè acaben en ?. Una paraula és una seqüència no buida de lletres i/o dígit. En les comandes fem servir {<paraula1> <paraula2> ... <paraulaN>} per referir-nos al conjunt de paraules format per <paraula1>, <paraula2>... i <paraulaN>. En canvi, amb "<paraula1> <paraula2> ...<paraulaN>" ens referim a la seqüència <paraula1>, <paraula2>, ... i <paraulaN>. Els caràcters "... " o {...} formen part de la comanda.

Les comandes s'exposen a continuació.

2 Comandes

2.1 Afegir, eliminar i transformar textos

La comanda

```
afegir text "<títol>"
autor "<autor>"
<contingut>
****
```

afegeix un text al sistema amb un títol, un autor i un contingut. Concretament, el títol està format per una o més frases amb la particularitat que l'última frase no té per què acabar amb punt, signe d'exclamació o interrogant. L'autor és una seqüència de paraules sense signes de puntuació; aquesta seqüència conté nom i cognoms d'una sola persona. Per exemple,

```
afegir text "La ciencia. Su metodo y su filosofia"
autor "Mario Bunge"
Por medio de la investigacion cientifica, el hombre ha
alcanzado una reconstruccion conceptual del mundo que
es cada vez mas amplia, profunda y exacta.
****
```

o bé

```
afegir text "The Picture of Dorian Gray"
autor "Oscar Wilde"
Those who find ugly meanings in beautiful things are corrupt
without being charming. This is a fault. Those who find
beautiful meanings in beautiful things are the cultivated.
For these there is hope. They are the elect to whom beautiful
things mean only Beauty. There is no such thing as a moral or
an immoral book. Books are well written, or badly written.
That is all.
****
```

La comanda

```
triar text {<paraula1> <paraula2>... <paraulaN>}
```

permet seleccionar l'únic text que conté les paraules <paraula1>, <paraula2>, ... <paraulaN> per a determinades operacions. Les paraules poden aparèixer

tant en el títol, com en l'autor, com en el contingut. Si hi ha més d'un text que les contingui o no n'hi ha cap, es considera que no es compleixen els requeriments de la comanda i, en aquest cas, s'ha de produir un missatge d'error tal com s'indica en la secció 2.5. Per exemple, si s'han executat les comandes `afegir text` de més amunt, aleshores la comanda

```
triar text {Wilde Beauty}
```

selecciona `The Picture of Dorian Gray`. En general, ens referirem a l'últim text seleccionat amb la comanda `triar text` amb l'expressió *últim text triat*.

La comanda

```
eliminar text
```

elimina del sistema l'últim text triat. **A partir d'aquest moment i fins la propera execució correcta de la comanda `triar text`, es considera que no existeix un últim text triat.**

La comanda

```
substitueix "<paraula1>" per "<paraula2>"
```

canvia totes les aparicions de `<paraula1>` per `<paraula2>` **en el contingut** de l'últim text triat.

2.2 Consultar informació en textos

La comanda

```
textos autor "<autor>" ?
```

mostra tots els textos d'un determinat autor (només el títol de cada text) ordenats alfabèticament per títol.

La comanda

```
tots textos ?
```

mostra tots els textos emmagatzemats en el sistema (només el títol i l'autor de cada text) ordenats per autor i després per títol.

La comanda

```
tots autors ?
```

mostra tots els autors indicant per a cadascun el nombre de textos que hi consten, el nombre total de frases i de paraules **del contingut** dels seus textos (els autors apareixen ordenats alfabèticament).

La comanda

`info ?`

mostra informació de l'últim text triat (autor, títol, nombre de frases, nombre de paraules i cites associades).

Consultar la informació dels textos:

`autor ?`

mostra l'autor de l'últim text triat.

La comanda

`contingut ?`

mostra el contingut de l'últim text triat amb les seves frases numerades a partir de l'1. Per mostrar una frase, s'escriurà el número de frase seguit d'un espai i de la frase, amb un únic salt de línia al final d'aquesta. En el format de sortida, els signes de puntuació aniran enganxats a la paraula prèvia i no hi haurà ni tabuladors ni dos espais seguits. Per exemple, si s'ha executat prèviament

`triar text {Picture Dorian}`

el resultat de `contingut ?` és

```
1 Those who find ugly meanings in beautiful things are corrupt
without being charming.
2 This is a fault.
3 Those who find beautiful meanings in beautiful things
are the cultivated.
4 For these there is hope.
5 They are the elect to whom beautiful things mean only Beauty.
6 There is no such thing as a moral or an immoral book.
7 Books are well written, or badly written.
8 That is all.
```

(Per restriccions de marges, algunes de les frases han ocupat dues línies)

La comanda

frases x y ?

mostra les frases entre la x -èsima i la y -èsima del contingut de l'últim text triat, extrems inclosos (si $x = y$, no es repeteix la frase). Aquesta comanda requereix que $1 \leq x \leq y \leq n$, on n és el nombre total de frases. Per exemple, sobre el mateix d'Oscar Wilde, el resultat de **frases 2 3** és

2 This is a fault.

3 Those who find beautiful meanings in beautiful things
are the cultivated.

La comanda

nombre de frases ?

mostra el nombre de frases del contingut de l'últim text triat.

nombre de paraules ?

en mostra el nombre de paraules.

La comanda

taula de freqüències ?

mostra totes les paraules del contingut del text ordenades decreixentment per freqüència (per a cada paraula la cadena que la representa i la seva freqüència) de l'últim text triat. En cas d'empat de freqüència les paraules s'ordenen creixentment, primer per llargada i després alfabèticament.

També podrem consultar les frases que compleixen una expressió booleana de paraules. Formalment, una *expressió booleana de paraules* és:

1. un conjunt de paraules
2. una expressió de la forma $(e_1 \ \& \ e_2)$ o bé $(e_1 \ | \ e_2)$

on e_1 i e_2 són expressions booleanes de paraules.

La comanda

frases <expressió> ?

mostra les frases del contingut de l'últim text triat que compleixen l'expressió (per a cada frase se'n mostra el seu número i el seu contingut, ordenades creixentment pel seu número). Per exemple,

`frases (({mel mato} | {fruita}) & {restaurant}) ?`

mostra les frases que contenen `restaurant` i, a més, sia `mel` i `mato`, sia `fruita`.

D'altra banda,

`frases ({casa taula} | {aigua}) ?`

o

`frases (({casa} & {taula}) | {aigua}) ?`

són equivalents i mostren les frases que contenen sia `casa` i `taula`, sia `aigua` en el contingut de l'últim text triat. En general, entre una subexpressió i una connectiva `&` o `|` hi pot haver un o més espais.

La comanda

`frases "<paraula1> <paraula2> ... <paraulaN>" ?`

mostra les frases on hi apareix (consecutivament, obviant els signes de puntuació) la seqüència de paraules `<paraula1>`, `<paraula2>` ... `<paraulaN>` en el contingut de l'últim text triat. Per exemple,

`frases "casa" ?`

mostra les frases on hi apareix la paraula `casa` mentre que

`frases "la casa" ?`

ho fa per la seqüència `la casa` però no, per exemple, per la seqüència `La casa`.

2.3 Cites

El sistema permet definir cites a partir del contingut d'un text.

`afegir cita x y`

afegeix una cita al sistema basada en les frases que van de la x -èsima a la y -èsima ($x \leq y$) del contingut de l'últim text triat. Aquesta comanda requereix que $1 \leq x \leq y \leq n$, on n és el nombre total de frases. No es pot afegir una cita que afecti al mateix text i frases que una cita existent, és a dir, no es permeten dues cites idèntiques (amb les mateixes frases) del mateix text. No obstant, es permet que dues cites comparteixin frases parcialment.

El sistema assigna a cada cita una referència que consisteix en les inicials de l'autor en majúscula seguides d'un identificador numèric comú a totes les referències amb les mateixes inicials. L'identificador numèric serà sempre el següent al nombre més alt associat mai (des de l'inici de l'execució del sistema) a les mateixes inicials, excepte quan el sistema encara no té cap cita amb les mateixes inicials, en el qual cas serà 1. Per tant, eliminar una cita no afecta al número que s'assignarà en un futur a una cita amb les mateixes inicials. A banda, si dos autors comparteixen inicials, compartiran identificador. Per exemple, si la primera cita que s'afegeix al sistema és de l'autor *Enric Fossas*, la referència serà EF1. Si la següent cita que afegim és de l'autor *Eudald Florit*, la seva referència serà EF2.

La comanda

`info cita "<referència>" ?`

mostra informació d'una cita (autor, títol, número de la frase inicial i número de la frase final, contingut de la frase o frases que la componen).

`cites autor "<autor>" ?`

mostra totes les cites d'un determinat autor (per a cada cita se'n mostra la seva referència, el contingut de les frases i títol del text d'on provenen) ordenades per referència.

`cites ?`

mostra totes les cites de l'últim text triat amb la comanda `triar text` (per a cada cita se'n mostra la seva referència, el contingut de les frases, autor i títol del text d'on provenen) ordenades per referència.

`totes cites ?`

mostra totes les cites emmagatzemades en el sistema (per a cada cita se'n mostra la seva referència, el contingut de les frases, autor i títol del text d'on provenen) ordenades per referència.

La comanda

`eliminar cita "<referència>"`

elimina la cita amb codi `<referència>` del sistema. Aquesta és l'única manera d'eliminar una cita; en particular, eliminar el text que conté la cita no té repercussions sobre les cites que se n'hagin pogut extreure.

De la mateixa manera, l'operació `substitueix` no afecta les cites del text on s'ha aplicat.

2.4 Acabament

L'execució del programa acaba quan s'introdueix la comanda

`sortir`

2.5 Errors

En el cas que una comanda no respecti els seus requeriments (és a dir, la informació que necessita l'operació), el sistema escriurà

`error`

Un exemple seria executar `triar text` amb un conjunt de paraules que no es troba en el sistema (perquè no ha estat introduït o perquè ha estat esborrat) o bé fer `info ?` sense haver triat cap text abans. Altres situacions, per exemple si falten paràmetres de l'operació en l'entrada (per exemple, en la comanda `afegir text`, ometre l'autor) o la comanda no s'escriu correctament (per exemple `afexir test` en lloc de `afegir text`), no les heu de controlar. Heu de suposar que aquestes altres situacions no es donaran mai.

3 Annex (implementació)

1. El format exacte de l'entrada i sortida es mostrarà en els jocs de proves públics.
2. Per processar l'entrada us serà útil llegir línies amb la comanda `getline` i processar el seu contingut mitjançant la classe `istringstream`. Un exemple de lectura d'una línia i escriptura de les paraules que la formen en línies diferents seria:

```
#include <sstream>
#include <string>
#include <iostream>
using namespace std;

int main(){
    string linia, m;
    getline(cin, linia);
    istringstream iss(linia);
    while (iss >> m) cout << m << endl;
}
```


3. Cada comanda genera sempre la primera línia de la mateixa comanda a l'inici, fins i tot si no genera cap sortida.
4. No es donaran més missatges d'error que els descrits a l'enunciat en les condicions indicades.
5. Podeu fer servir qualsevol estructura de dades vista a la teoria o al laboratori.
6. Caldrà donar prioritat a l'eficiència de les operacions de consulta per sobre d'insercions, modificacions o esborraments.
7. Volem que les operacions **tots autors ?** i **taula de freqüències** siguin especialment ràpides.
8. L'algorisme per processar les expressions algebraiques que combinen **&** i **|** ha de ser recursiu.
9. L'entrada del programa i, en particular, el contingut del text han d'estar formats per caràcters en codi ASCII més petit que 128 (això exclou accents i determinats signes de puntuació).
10. Els jocs de proves senzills (amb dades petites) no permeten detectar problemes d'eficiència. Recomanem l'ús de textos de mida realista, com els textos aleatoris que posarem a la vostra disposició a la pàgina web de la pràctica.