



## Ingénierie logicielle en équipe

### Introduction

Premièrement, 30 minutes pour faire le design ensemble. Ensuite, vous codez seul pendant 2 heures. Finalement, 30 minutes pour réunir le tour. Vous aurez besoin d'un très bon design pour tout faire fonctionner!

### Competition

Votre tâche est de faire un moteur de recherche qui reçoit une query et retourne toutes les phrases de sa base de donnée qui correspondent à cette query.

### Entrée/sortie attendue

#### Entrée

Une Command Line Interface qui prend le fichier de phrases comme premier argument et la query comme second argument.

On obtient donc quelque chose du genre : `./program sentences.txt "this is a query"`

#### Sortie

Les résultats écrit dans la console

Vous trouverez une base pour python et java qui répond à ces besoins. Votre code va dans `search_engine.py` / `SearchEngine.java`. Vous pouvez éditer ce que vous voulez dans cette base au besoin, mais vous devez satisfaire les exigences I/O. Vous pouvez également utiliser n'importe quel langage si ces exigences sont satisfaites.

Vous trouverez aussi un exemple de fichier `sentences.txt` qui contient des phrases pour vos tests. Vous pouvez ajouter ce que vous voulez à ce fichier. Ce n'est pas le même fichier qui sera utilisé lors de la correction.

#### Exécuter en Java

Aller au dossier /bin généré par votre IDE. Exécuter cette commande :

```
java org.csgames.tse.Main ../sentences.txt "this is an \"example query\""
```

Si vous n'utilisez pas un IDE pour Java, vous devriez déjà savoir comment le compiler et l'exécuter.

## Exécuter en Python

Aller au dossier contenant main.py. Exécuter cette commande :

```
python main.py sentences.txt "this is an \"example query\""
```

## Queries

### Notes

- [ ] dans les explications sont utilisés pour distinguer les explications des mots des query/phrases.
- Ce ne sont pas les queries qui seront utilisé pour la correction: les mots ne sont pas les mêmes, les nombres de mots dans les exact match ne sont pas les mêmes, des opérateurs CALC différents, etc.
- Les résultats ne sont pas sensibles à la casse (voir query #1), mais vous pouvez considérer que les mot-clés AND, OR, NOT, CALC vont toujours être en MAJUSCULES dans le query.
- Lors des recherches, on cherche le mot exact et complet (voir la query #1). Il n'y a pas de ponctuation à la fin des mots. Si il y a, par exemple, un point d'interrogation à la fin d'une phrase, il sera séparé par un espace ([Hello ?], not [Hello!]).
- La wild card (\*) remplace 0 caractères ou plus (see query #7)
- La wild card (\*) n'a pas le même effet dépendant du contexte (mélangé à un mot, entre des guillemets, etc.)

### Format

1ère ligne : Devrait être assez clair...

2e ligne : Description de la Query: exemple de query

3e ligne : Ce que la query devrait retourner

4e ligne : Précisions

## Queries simples

#1 (2 pts)

Simple word : example

Retourne les phrases qui contiennent le mot [example] (**mot exact**, ne devrait pas trouver [exam]).

Precisions : rappel... **INSENSIBLE À LA CASSE**. [Example], [example], [exAmPLe].. ça ne devrait pas faire de différence.

#2 (2 queries, 2 pts each)

Exact match (quotes) : "an example"

Retourne les phrases qui contiennent la phrase exacte [an example].

#3 (3 pts)

AND : an example

Retourne les phrases qui contiennent les 2 mots, n'importe où dans la phrase.

Precisions : AND est implicite

#4 (3 pts)

OR : an OR example

Retourne les phrases qui contiennent un des deux mots.

#5 (3 pts)

NOT : NOT example

Retourne les phrases qui ne contiennent pas [example].

#6 (2 pts)

Wild card : \*

Retourne tout.

#7 (4 pts)

Wild card in exact match : "this \* example"

Retourne les phrases correspondant à la séquence suivante : [this], suivi de n'importe quoi, puis finit par [example].

Precisions : \* remplace 0 caractères ou plus (la séquence peut être [this example]).

#8 (3 pts)

Wild card à la fin d'un mot : exam\*

Retourne les phrases avec des mots qui commencent avec [exam].

#9 (3 pts)

Wild card au début d'un mot : \*ample

Retourne les phrases avec des mots qui finissent par [ample].

#10 (2 pts)

Wild card au début et à la fin d'un mot: \*test\*

Retourne les phrases qui contiennent [test].

#11 (2 queries, 3 pts each)

CALC : {CALC 5+5\*5}

Retourne les phrases qui contiennent 30

Precisions : Les "brackets" sont toujours là, et il n'y a aucun espace entre les nombres. Vous devez considérer la priorité des opérations et des parenthèses.

#12 (3 pts)

Range : 67..90

Retourne les phrases qui contiennent des nombres entre 67 et 90

Precisions : Noter que c'est **DEUX** points entre 67 et 90.

#13 (3 pts)

Multiple occurrence : example{2}

Retourne les phrases qui contiennent le mot [example] **AU MOINS** deux fois (le nombre dans les "brackets")

## Mixed queries

#14 (4 pts)

Simple word + AND + Exact match : hello "an example"

Retourne les phrases qui contiennent [hello] ET [an example]

#15 (4 pts)

Simple word + AND + NOT : hello NOT example

Retourne les phrases qui contiennent [hello] et pas [example]

#16 (4 pts)

Simple word + AND + OR : hello test OR example

Retourne les phrases qui contiennent [hello] ET [test], OU seulement [example]

#17 (4 pts)

Exact match + CALC : "There are {CALC 34+8} ninjas in this query"

Retourne la même chose que si le query était: "There are 42 ninjas in this query"

#18 (4 pts)

Range + AND + simple word : 10..100 beers

Retourne les phrases qui contiennent un nombre entre 10 et 100 ET le mot [beers].

#19 (4 pts)

Range + CALC : 42..{CALC 9\*(30+30)}

Pareil que CALC 42..69

#20 (4 pts)

Exact match + multiple occurrences : "an example"{3}

Retourne les phrases qui contiennent [an example] au moins 3 fois

#21 (4 pts)

Simple word + AND + multiple occurrences : an{2} example

Retourne les phrases qui contiennent [an] au moins deux fois ET [example] au moins une fois.

#22 (4 pts)

Simple word + OR + multiple occurrences : an{3} OR example

Retourne les phrases qui contiennent [an] au moins trois fois OU example au moins une fois

#23 (4 pts)

Range + OR : 15..25 OR 58..93

Retourne les phrases qui contiennent un nombre en 15 et 25 OU un nombre entre 58 et 93

#24 (4 pts)

Wild card in words and exact match : "this \* exam\*"

Retourne les phrases avec la séquence suivante : [this], suivi par n'importe quoi, suivi par un mot qui commence avec [exam].

#25 (15 pts)

Query du plaisir :

"the \* example"{4} {CALC 8\*(8+9/3)}..{CALC 90000/10+1} OR \*park NOT bench