



Piscine de Synthèse
Tech 2
2015

autoCompletion
Moteur d'Auto-Complétion
pour GPS





Formalités de rendu

nom du binaire: autoCompletion

nom du dépôt: autoCompletion

droits sur le dépôt: ramassage-tek

langage: C, C++, perl 5, python 2.7, ruby 2.1, php 5, bash 4

compilation (si nécessaire): via Makefile, avec les règles re, clean, fclean

taille de groupe: 1

début du projet: 1^{er} jour, 9h00

checkpoint: 5^{ème} jour, après-midi

date de rendu: 5^{ème} jour, 23h42



Sujet

Depuis l'accident de voiture de l'oncle Edmond alors qu'il réglait son GPS, la société Fin-YourWay.com a décidé de revoir le processus de saisie d'adresse.

Toujours alerte sur l'actualité des entreprises du numérique, vous décidez de leur proposer un POC d'un algorithme d'auto-complétion d'adresse.

Voici le principe de votre algorithme :

1. le format d'adresse est : *ville, numéro typeDeVoie nomDeLaVoie*.

typeDeVoie pouvant prendre les valeurs :

- *allée*,
- *avenue*,
- *boulevard*,
- *chemin*,
- *impasse*,
- *place*,
- *quai*,
- *rue*,
- *square*.

2. l'auto-complétion se fait lettre par lettre, en proposant les 5 lettres les plus probables, par ordre de probabilité décroissante,

3. l'auto-complétion se fait d'abord sur la ville, puis sur le nom de la voie,

4. l'auto-complétion peut se faire sur n'importe quelle partie du nom de ville ou de voie (La Plaine Saint Denis comporte par exemple 4 parties),

5. s'il ne reste qu'une seule possibilité, l'adresse complète de la ville s'affiche,

6. une fois un mot terminé, la liste des adresses contenant ce mot s'affiche.





Dictionnaire

Votre programme prend en paramètre un dictionnaire d'adresses contenant une adresse par ligne, formatée de la même manière que ci-dessus.

Les majuscules et les minuscules sont ignorées.
Les tirets et apostrophes sont aussi ignorés.



Les lignes mal formatées sont affichées sur la sortie d'erreur.

Correction d'erreur

Une fois cette partie fonctionnelle, implémentez un algorithme de correction d'erreur qui rectifie les adresses mal formatées (les exemples ci-dessous ne prennent pas en compte cette correction d'erreur).

Bien évidemment, seules les adresses contenant la totalité des informations peuvent être reformatées.



Un bon parseur semble s'imposer... d'autant que la grammaire associée est simplissime...



La libC est AUTORISEE pour le projet, exceptée la fonction *system*.





Prototypage

```
Terminal
~/PiscineSynthese> ./autoCompletion dictionnaire
```

Le programme affiche sur la sortie standard les 5 lettres les plus probables pour le début du nom de la ville (ou d'un mot contenu dans le nom de ville) et se met en attente sur l'entrée standard de la lettre suivante.

Ensuite, :

si le mot n'est pas terminé

- si une seule lettre est possible, elle est ajoutée au mot en cours, et le processus est réitéré,
- sinon, les 5 lettres suivantes les plus probables sont proposées, et le programme se met en attente d'un caractère,

si le mot est terminé et que la ville est inconnue,

- s'il ne correspond qu'à une seule ville, le nom de cette ville est affichée, suivi de ', ' puis les 5 lettres les plus probables pour le début du nom de voie (ou pour le début d'un mot contenu dans le nom de voie) sont affichées, et le programme se met en attente d'un caractère,
- s'il correspond à plusieurs noms de ville, ils sont tous affichés (par ordre alphabétique) associés à un numéro. Le programme se met alors en attente d'un nombre ou d'un caractère,

si le mot est terminé et que la ville est connue,

- s'il correspond à un seul nom de voie et à une seule adresse, elle est affichée, et le programme s'arrête,
- s'il correspond à un seul nom de voie et plusieurs adresses, elles sont toutes affichées (par ordre alphabétique) associées à un numéro et le programme se met en attente d'un nombre,
- s'il correspond à plusieurs noms de voie, ils sont tous affichés (par ordre alphabétique) associés à un numéro. Le programme se met alors en attente d'un nombre ou d'un caractère.

A tout moment la lecture du mot-clef *ABORT* quitte le programme.



En cas de lettres équiprobables, elles seront affichées par ordre alphanumérique.





Les deux seules erreurs à afficher sont *Argument invalide* et *Adresse inexistante*.



Tous les messages d'erreur sont à afficher sur la sortie d'erreur !





Exemples



La sortie de votre programme devra être rigoureusement identique à celle des exemples ci-dessous.



Les lettres validées sont affichées en minuscules.
Celles proposées sont en majuscules.

```
Terminal
~/PiscineSynthese>cat miniDico
Paris, 458 boulevard Saint-Germain
Paris, 343 boulevard Saint-Germain
Marseille, 343 boulevard Camille Flammarion
Marseille, 29 rue Camille Desmoulins
Marseille, 1 chemin des Aubagnens
Paris, 12 rue des singes
Paris, 34 quai VoLtAiRe
Paris, 34 rue Voltaire
Lille, 120 boulevard Victor Hugo
Marseille, 50 rue Voltaire
Toulouse, 90 rue Voltaire
Strasbourg 84 rue du Bouclier
Marseille, 78 boulevard de la libération
Lille, 30 rue Victor Danel
Mont Saint Martin, 42 rue de Bordeaux
Mont de Marsan, 100 avenue Pierre de Coubertin
Lyon, 56 rue du Docteur Albéric Pont
Lyon, rue du Docteur Albéric Pont
56 rue du Docteur Albéric Pont, Lyon
Lyon 56 grande rue
Lille, 90 rue d'Arras
Lille, 76 impasse Georges Pompidou
Lyon, 2 allée des fleurs
```

```
Terminal
~/PiscineSynthese>echo ABORT | ./autoCompletion miniDico 1>\dev\>null
Strasbourg 84 rue du Bouclier
Lyon, rue du Docteur Albéric Pont
56 rue du Docteur Albéric Pont, Lyon
Lyon 56 grande rue
```





Terminal

```
~/PiscineSynthese>echo x | ./autoCompletion miniDico
Strasbourg 84 rue du Bouclier
Lyon, rue du Docteur Albéric Pont
56 rue du Docteur Albéric Pont, Lyon
Lyon 56 grande rue
{m} {l} {p} {s} {d}
Adresse inexistante
```

Terminal

```
~/PiscineSynthese>cat test1
L
ABORT
~/PiscineSynthese>./autoCompletion miniDico 2>\dev\null < test1
{m} {l} {p} {s} {d}
{Li} {Ly}
```

Terminal

```
~/PiscineSynthese>cat test2
l
i
v
H

~/PiscineSynthese>./autoCompletion miniDico 2>\dev\null < test2
{m} {l} {p} {s} {d}
{Li} {Ly}
{LILLE, d} {LILLE, v} {LILLE, a} {LILLE, g} {LILLE, h}
{LILLE, VICTOR d} {LILLE, VICTOR h}
=> Lille, 120 boulevard Victor Hugo
```

Terminal

```
~/PiscineSynthese>cat test3
p
S
a
2

~/PiscineSynthese>./autoCompletion miniDico 2>\dev\null < test3
{m} {l} {p} {s} {d}
{PARIS, s} {PARIS, g} {PARIS, v} {PARIS, d}
{PARIS, Sa} {PARIS, Si}
{1 : PARIS, 343 BOULEVARD SAINT-GERMAIN} {2 : PARIS, 458 BOULEVARD SAINT-GERMAIN}
=> Paris, 458 boulevard Saint-Germain
```





Terminal

```
~/PiscineSynthese>cat test4  
D
```

```
~/PiscineSynthese>./autoCompletion miniDico 2>\dev\null < test4  
{m} {l} {p} {s} {d}  
=> Mont de Marsan, 100 avenue Pierre de Coubertin
```

Terminal

```
~/PiscineSynthese>cat test5  
p  
v  
1
```

```
~/PiscineSynthese>./autoCompletion miniDico 2>\dev\null < test5  
{m} {l} {p} {s} {d}  
{PARIS, s} {PARIS, g} {PARIS, v} {PARIS, d}  
{1 : PARIS, 34 quai VOLTAIRE} {2 : PARIS, 34 rue VOLTAIRE}  
=> Paris, 34 quai VoLtAiRe
```

Terminal

```
~/PiscineSynthese>cat test6  
l  
i  
P
```

```
~/PiscineSynthese>./autoCompletion miniDico 2>\dev\null < test6  
{m} {l} {p} {s} {d}  
{Li} {Ly}  
{LILLE, d} {LILLE, v} {LILLE, a} {LILLE, g} {LILLE, h}  
=> Lille, 76 impasse Georges Pompidou
```

Terminal

```
~/PiscineSynthese>cat test7  
l  
i  
d  
A  
R
```

```
~/PiscineSynthese>./autoCompletion miniDico 2>\dev\null < test7  
{m} {l} {p} {s} {d}  
{Li} {Ly}  
{LILLE, d} {LILLE, v} {LILLE, a} {LILLE, g} {LILLE, h}  
{LILLE, D'Ar} {LILLE, DAn}  
=> Lille, 90 rue d'Arras
```





```
Terminal - + X
~/PiscineSynthese>cat test8
m
A
d
E
S
M

~/PiscineSynthese>./autoCompletion miniDico 2>\dev\null < test7
{m} {l} {p} {s} {d}
{Ma} {Mo}
{MARSEILLE, d} {MARSEILLE, c} {MARSEILLE, l} {MARSEILLE, a} {MARSEILLE, f}
{1 : MARSEILLE, 78 boulevard DE la libération}
{1 : MARSEILLE, 1 chemin DES aubagnens}
=> Marseille, 29 rue Camille Desmoulins
```

