



# Rush d'Algo

## HotRace

42 staff [pedago@42.fr](mailto:pedago@42.fr)

*Summary: Ce rush est un concours de rapidité pour vous faire réfléchir sur des algorithmes et techniques de programmation efficaces. Il nous permettra de sélectionner une équipe de 10 pour participer au challenge Codin Game. Les sélectionnés qui participeront au challenge pourront gagner des cadeaux.*

# Contents

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Sujet</b>	<b>3</b>
II.1	Ce-quoi-donc-qui-faut-il-faire . . . . .	3
II.2	Rendu . . . . .	5
II.3	Fonctions autorisées . . . . .	6
<b>III</b>	<b>Consignes</b>	<b>7</b>

# Chapter I

## Préambule

En travers de l'histoire, l'être humain s'est peu à peu affranchi des servitudes naturelles qui entravaient le plein épanouissement de son caractère fondamental, et le développement de son identité culturelle fondamentale. Ainsi, en l'an 1524 de notre ère, fut inventé un écriteau robuste et courageux, lui-même fixé sur un support robuste et courageux, portant ces mots : *“Ne fermez pas la porte, le blount s'en chargera.”*



Figure I.1: L'écriteau robuste et courageux

# Chapter II

## Sujet

### II.1 Ce-quoi-donc-qui-faut-il-faire

Finalement, c'est pas très compliqué de faire un moteur de recherche comme *Google*. Il suffit d'avoir des tas d'infos, puis de simplement chercher des mots clefs à l'intérieur. Non, la difficulté, c'est de faire cela rapidement. Et alors là, il y a une lutte acharnée entre les *google*, *yahoo*, *bing*, et tout un tas d'autres qui n'existaient presque déjà plus quand vous êtes nés. C'est une course où tout le monde donne le meilleur : la **HotRace**.

Participez-donc à la HotRace. Pour cela, il vous suffit de réaliser un programme qui va émuler un moteur de recherche. Il pourra emmagasiner beaucoup d'informations, les modifier, les supprimer et accessoirement les retourner à l'utilisateur.

- Votre programme s'appelle **hotrace**.
- Votre programme ne prend aucun parametre.
- Votre programme lit sur l'entrée standard les informations et les éléments à rechercher.
- Votre programme supprime les données d'un mot-clef précédé d'un "!"
- Votre programme affiche la valeur correspondant à chaque recherche suivie d'un "\n".
- Si nous passons un mot clé sans valeur, c'est que nous voulons afficher sa valeur (si elle existe).
- Quand il y a un " ! ", cela doit supprimer la pair <clé,valeur> de la db.
- Le retour doit coller exactement à l'exemple.
- Si la ligne contient deux fois égal, le deuxième fait partie de la valeur.
- Il n'y aura pas plus de caractère spéciaux que ! en premier caractère et un premier égal.
- Si clé existe déjà, l'input est ignoré.
- Il n'y aura que des caractères ascii imprimables.

- Vous n'aurez seulement le droit à eof / sigint pour quitter le programme (il n'y a pas besoin de bien free).

Le format des infos sur l'entrée standard est le suivant :

```
mot-clef-1=valeur1
mot-clef-2=valeur2
[...]
mot-clef-x=valeurx
!mot-clef-2
!mauvaise-clef
mot clef avec espace=valeur avec espace
mot-clef-recherche1
mot-clef-recherche2
[...]
```

L'exemple donne des input avec des output correspondant. ex: Motcle=valeur en input rentre une paire dans le dictionnaire, sans output. Motcle en input donne la valeur en output ... Si la clé existe déjà, cela ne rentre rien. La recherche peut ne rien donner, alors le message suivant s'affiche :

```
mot-clef-recherche: Not found
```

Remplacez bien sur "mot-clef-recherche" par le mot clef qui a été effectivement recherché. Note: Effacer un mot clef ne renvoie aucun message, même s'il n'existe pas.

Parmi les programmes qui fonctionnent correctement, un classement de vitesse sera fait. Votre but ? Être le plus rapide. La vitesse, et donc l'optimisation de votre programme, sont cruciaux. Attention dans les virages.

Ce rush se fera seul.

P.S. : si vous pensez que c'est compliqué, attendez d'avoir à le faire en script shell.

## II.2 Rendu

- Le rendu se fera sur le github de 42 Paris. Vous devrez faire des PR dans le dossier "rendu" : <https://github.com/42Paris/Challenge-HotRace> . Vous devrez créer un dossier content votre login, et uniquement votre login. Vous rendrez tout à la racine de ce dossier. Vous rendrez aussi un fichier `auteur` contenant votre login sur une ligne et suivi d'un `'\n'`:

```
$>cat -e auteur  
xlogin$  
ylogin$  
$>
```

- Vous devez avoir un `Makefile` avec les règles usuelles. Dans le doute, mettez toutes les règles connues.
- Seul le contenu présent sur votre dépôt sera évalué.

## II.3 Fonctions autorisées

- Toute fonction standard qui ne fait pas partie d'une classe

# Chapter III

## Consignes

- Vous pouvez faire le projet en C, C++ ou Rust.
- Vous n'avez pas le droit aux containers standard de ces langages pour vous aider. Si vous en avez besoin : réimplémentez les.
- Vous devez gérer les erreurs de façon sensible. En aucun cas votre programme ne doit quitter de façon inattendue (Segmentation fault, bus error, double free, etc).
- Toute mémoire allouée sur le tas doit être libérée proprement.

Bon courage à tous pour ce rush !