



Rush du module Algorithmie I

HotRace

42 staff staff@42.fr

Résumé: Ce rush est un concours de rapidité pour vous faire réfléchir sur des algorithmes et techniques de programmation efficaces.

Table des matières

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

Chapitre 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

Chapitre 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 a le feu aux fesses : la **HotRace**.

Participez-donc à la HotRace. Pour cela, il vous suffit de réaliser un programme qui va fonctionner en deux temps : une première étape consiste à emmagasiner des informations de type "mot-clef = valeur", beaucoup d'information, et la seconde étape est une demande de recherche dans ces informations, en transmettant un mot-clef.

- Votre programme s'appelle **hotrace**.
- Votre programme ne prend aucun parametre.
- Votre programme lit sur l'entrée standard les informations.
- Votre programme lit sur l'entrée standard les éléments à rechercher.
- Votre programme affiche la valeur correspondant à chaque recherche suivie d'un "\n".

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

```
mot-clef-1
valeur1
mot-clef-2
valeur2
[...]
mot-clef-x
valeursx

mot-clef-recherche1
mot-clef-recherche2
[...]
```

Notez le saut de ligne qui sépare les informations de la recherche.

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.

Parmi les programmes qui fonctionnent correctement, le plus rapide a 20, les 2 suivants 19, les 4 suivants 18, et ainsi de suite. La vitesse, et donc l’optimisation de votre programme, sont cruciaux.

Ce rush se fait à 2.

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

II.2 Rendu

- Vous devez rendre, à la racine de votre dépôt de rendu, un fichier **auteur** contenant vos 2 login chacun 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

- read
- write
- malloc
- free
- strerror
- la directive de compilation `__asm__`

Chapitre III

Consignes

- La note sera obtenue par la moulinette de rapidité, si la correction s'est parfaitement passée.
- Votre projet doit être à la Norme. La Norminette ne sera pas utilisée pour vérifier la Norme qui s'applique donc dans son ensemble.
- 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.
- Vous ne pouvez pas utiliser votre bibliothèque `libft`.

Bon courage à tous pour ce rush !