DAEMONLAB

READLINE

Lire une ligne entière

- DaemonLab -

Readline est une fonction servant à lire une ligne entière.
Cette fonction est extrêmement utile, au moins autant que d'autres fonctions tel que ***split****.*

INDEX

01 – Avant-propos

Votre travail doit être rendu via votre bibliothèque, **libstd**.

Si vous faites erreur et que le dossier que vous utilisez pour votre rendu est différent, vous ne serez pas évalué faute d'avoir pu trouver votre travail.

Ce travail est a effectuer seul. Vous pouvez bien sur échanger avec vos camarades, néanmoins vous devez être l'auteur de votre travail. Utiliser le code d'un autre, c'est **tricher**. Et tricher annule **toutes** les médailles que vous avez reçu sur l'activité. La vérification de la triche est réalisée de la même manière que la correction : de manière **automatique**. Prenez garde si vous pensez pouvoir passer au travers.

Votre rendu doit respecter **strictement** l'ensemble des règles suivantes :

- Il ne doit contenir **aucun** fichier objet. (*.o)
- Il ne doit contenir **aucun** fichier tampon. (*~, #*#)
- Il ne doit pas contenir votre production finale (programme ou bibliothèque)

La présence d'un fichier interdit mettra immédiatement fin à votre évaluation.

Votre programme doit respecter les Tables de la Norme dans leur intégralité. Vous êtes invité à les observer depuis **l'Infosphère**. Elles sont disponibles comme ressource de cette activité.

02 – Fonctions autorisées

La bibliothèque logicielle venant avec le C est vaste et disponible. La LibLapin, que vous utilisez dans vos projets multimédia, est également vaste… Cependant nous avons fait le choix de vous interdire leurs utilisation intégrales, afin de vous amener progressivement à reprogrammer vous même ses fonctionnalités les plus utiles.

L'utilisation d'une fonction interdite est assimilée à de la triche. La triche provoque l'arrêt de l'évaluation et la perte des médailles.

Vous n'avez le droit d'utiliser aucune fonction issue de la LibC ou de la LibLapin à l'exception de celles que nous vous autoriserons explicitement.

Pouvoir utiliser une fonction ne signifie pas nécessairement que celle-ci soit utile à votre cas.

Pour cette activité, issu de la LibC, vous n'avez le droit qu'à la liste suivante :

- open
- close
- read
- srand
- cos
- atan2
- time
- write
- alloca
- atexit
- rand
- sin
- sqrt
- malloc, free

Écrivez la fonction suivante :

```
void            *std_realloc(void            *ptr,
                             size_t          new_size);
```

Cette fonction alloue un nouvel espace mémoire faisant new_size octets et copie le contenu de ptr, de taille old_size dans le nouvel espace.

Elle vous sera utile pour le prochain exercice.

Pour cette fonction, vous aurez le droit (et l'obligation) d'employer la fonction malloc_usable_size.

Écrivez la fonction suivante :

```
char            *std_readline(int         fd);
```

Cette fonction effectue des lectures sur fd tant qu'une ligne complète n'a pas été récupérée. Une fois qu'une ligne a été récupérée, elle est renvoyée. Une ligne est complète si elle est terminée par un saut de ligne. La valeur de retour doit pouvoir être libérée par un appel à free.

La principale difficulté de l'exercice consiste à gérer les caractères excédentaires. Comme la fonction doit pouvoir être appelée plusieurs fois d'affilés, vous allez être obligé de vous arranger pour que la fonction conserve des informations sur ses lectures passées.

Comment faire ? En utilisant un espace mémoire de taille variable, à l'aide de std_realloc !

Une fois cette première étape terminée, à l'aide du mot-clef static, vous pourrez réaliser des read plus grand. Le mot clef static se place devant une variable de la façon suivante, par exemple :

```
static int i;
```

Une variable statique continue d'exister à la fin de la fonction, sa valeur est donc inchangée entre les appels. Vous pouvez donc utiliser une variable statique pour conserver l'état de votre lecture. Si vous n'initialisez pas votre variable dès le départ, tous ses octets seront par défaut à zéro, contrairement à une variable normale.

Lorsque vous utilisez une fonction statique, vous pouvez faire une entorse aux Tables de la norme et faire une assignation en ligne. Cette variable peut tout à fait être une structure ou un tableau. Ou une structure contenant un tableau.