D A E M O N L A B

Variadique

Fonctions à paramètres variables

- DaemonLab -

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

Pour information, va_start, va_arg et va_end sont des macros, vous y avez donc droit.

03 – Travail

Écrivez la fonction suivante :

```
int             std_vputs(size_t        nbr,
                          ...);
```

Cette fonction prend en paramètre le nombre de paramètres qu'elle prendra en plus à la suite de **nbr**. Ces paramètres seront tous des chaînes de caractères.

Pour réussir cet exercice, vous aurez besoin d'utiliser les macros, fonctions et symboles de la bibliothèque standard **stdarg.h**. Parmi ces symboles : **va_list**, **va_start**, **va_end** et **va_arg**. Le premier est le type qui permettra créer des variable pour « pointer » sur la liste de paramètres. **va_start** l'initialisera, **va_end** l'arrêtera et la libérera. **va_arg** permettra d'extraire un paramètre. L'utilisation de ces fonctions est détaillés dans leurs manuels respectifs.

Cette fonction renverra le nombre d'octets qui ont été écrit. Si une erreur a lieu lors de l'écriture, la fonction renverra -1.

Écrivez la fonction suivante :

```
int             std_mini_printf(const char     *pattern,
                                ...);
```

Cette fonction prend en paramètre une chaîne de caractère **pattern**. Cette chaîne de caractère sera affichée à l'exception de certains symboles situés à l'intérieur. Ces symboles représentent dans l'ordre les paramètres supposément passé à la fonction à la suite de **pattern**. Ces symboles sont **%** et **@**. Si le symbole **%** est trouvé, alors il faudra chercher un paramètre de type **int**. Si le symbole **@** est trouvé, alors cela sera un paramètre de type **const char ***.

La fonction renverra le nombre d'octets qui ont été écrit. Si une erreur a lieu lors de l'écriture, la fonction renverra -1.