DAEMONLAB

VSNPRINTF

Formatage de texte avancé

- DaemonLab -

INDEX

01 – Détails administratifs

Votre travail doit être rendu via votre bibliothèque personnelle, **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
- malloc
- write
- alloca
- atexit
- rand
- sin
- sqrt
- free

03 – Méthode de construction

Il peut vous être demandé d'écrire des programmes ou des fonctions.

Dans le cas des programmes, il vous sera toujours demandé de fournir un **dossier** pour l'exercice le requérant. Un **Makefile** vous sera également demandé. Le **nom du programme** de sortie vous sera précisé à chaque fois. Un Makefile incorrect, un mauvais nom de programme, et votre correction n'aura pas lieu…

Dans le cadre des fonctions, il vous ai demandé de fournir le fichier dans votre **dossier de bibliothèque personnelle**, de sorte à ce que vous puissiez utiliser toutes les fonctions que vous avez déjà réalisé jusqu'ici. Pour rappel, le dossier de votre bibliothèque doit être placé à la racine de votre espace personnel et s'appeler **libstd/.**

N'oubliez pas d'entretenir avec soin votre dossier **libstd/** de sorte à ce qu'il soit toujours propre, respecte la norme et soit en état de compiler… sans quoi elle fera obstacle à la correction.

Votre compilation devra toujours comporter les options **-W**, **-Wall** et **-Werror**.

Dans le cadre de la programmation multimédia, le système de correction établira toujours la variable d'environnement **BMALLOC** à 1. Si vous utilisez le modèle de projet, cela provoquera l'utilisation de **bunny_malloc** dans votre bibliothèque personnelle comme dans votre projet rendu.

04 – Projet

Votre travail consistera à implémenter la fonction suivante :

```
int             std_vsnprintf(char          *str,
                              size_t        size,
                              const char    *pattern,
                              va_list       lst);
```

Vous trouverez les informations dont vous avez besoin dans le manuel de **vsnprintf**. Vous **devez** implémenter au moins les formats **%d**, **%c** **%f** **%s**, **%p** et **%%**.Vous **devez** implémenter les méthodes de bourrage **espace** et **0**. Vous **devez** implémenter les options + et -.

Voici une étape intermédiaire pour vous aider :

```
bool            std_catchar(char            *str,
                            size_t          *i,
                            size_t          max,
                            char            c);
```

La fonction **catchar** met à la position ***i** dans **str** le caractère **c** si ***i** est inférieur à **max**. Elle place le terminateur nul **'\0'** intelligemment de manière à ce que la fin soit **toujours** présente à l'endroit logique et sécuritaire.

Si le caractère **c** n'a pas été ajouté, elle renvoi alors **faux** sinon elle renvoi **vrai**.

Si un caractère a été ajouté, alors ***i** est augmenté de 1.

Cette fonction remplace aisément **std_putchar**.

05 – Retombées

Utilisez **vnsprintf** pour implémenter, dans l'ordre, les fonctions suivantes :

```
int             std_vdprintf(int            fd,
                             const char     *pattern,
                             va_list        lst);

int             std_vprintf(const char      *pattern,
                            va_list         lst);

int             std_snprintf(char           *str,
                             size_t         size,
                             const char     *pattern,
                             ...);

int             std_dprintf(int             fd,
                            const char      *pattern,
                            ...);

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

Ces fonctions doivent exploiter votre fonction **std_vsnprintf** ou s'entre-exploiter pour leur fonctionnement. Vous ne **devez pas** copier coller de code : vous obtiendrez sinon des médailles négatives…

Bonus : réalisez les fonction suivantes, **asprintf** étant basé sur **vasprintf**.

```
int             std_vasprintf(char          **str,
                              const char    *pattern,
                              va_list       lst);
int             std_asprintf(char           **str,
                             const char     *pattern,
                             ...);
```