



D A E M O N L A B

Journée 7

Structures, pointeurs sur fonctions
Et votre bibliothèque

- DaemonLab -

Durant cette journée, vous découvrirez un élément fondamental du C : les structures de données ainsi qu'une nouvelle catégorie de pointeurs : ceux qui pointent non pas sur des données mais sur des fonctions. Nous allons également aborder la compilation de votre propre bibliothèque logicielle. Attention, la compilation sera différente désormais !

Ce document est strictement personnel et ne doit en aucun cas être diffusé.



INDEX

- 01 – Avant-propos
- 02 – Fonctions autorisées
- 03 – Fractions
- 04 – Stylo
- 05 – Votre bibliothèque



01 – Avant-propos

Votre travail doit être rendu via le dossier `~/hyperspace/j7/` dans votre espace personnel.

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 à effectuer seul. Vous pouvez bien sûr 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. Nous avons cependant fait le choix de vous interdire son utilisation, 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 à l'exception de celles que nous vous autoriserons explicitement.

Pour cette activité, vous avez le droit à l'appel système **write** ainsi qu'aux fonctions **malloc** et **free**.



O3 – Fractions

Mettre dans le fichier : `include/fraction.h`, `src/*.c`

Attention, désormais, vous placerez vos fichiers entêtes dans le dossier `include/` de votre rendu. Les fichiers `.c` seront placés dans `src/` et porteront comme l'indique la table des normes le **nom de leur fonction** (en dehors du préfixe). Exemple, la fonction `std_blob` sera dans `src/blob.c`

Attention, encore, à partir de maintenant, nous compilerons tous vos fichiers d'un coup. Cela signifie que vous ne devez **plus** fournir dans vos fichiers les dépendances à vos fonctions. Plus de `tc_putchar` traînant ici et là, plus de `std_strcpy` appelé par `std_strdup` dans le même fichier...

La table des normes impose la limite d'une seule fonction non-statique par fichier. Organisez votre code en fonction de cet impératif.

Définissez dans le fichier entête demandé la structure suivante et les fonctions associées :

```
typedef struct s_fraction
{
    int      numerator;
    int      denominator;
    t_fraction;
}

t_fraction  *new_fraction(int num, int denum);

double      fraction_compute(t_fraction *f);

t_fraction  *fraction_add(const t_fraction *a,
                           const t_fraction *b);
t_fraction  *fraction_sub(const t_fraction *a,
                           const t_fraction *b);
t_fraction  *fraction_mul(const t_fraction *a,
                           const t_fraction *b);
t_fraction  *fraction_div(const t_fraction *a,
                           const t_fraction *b);
void        fraction_simplify(t_fraction *f);
```

Écrivez ensuite chacune des fonctions dont les prototypes sont indiqués dans cet entête.

Les fonctions `add`, `sub`, `mul` et `div` génère à chaque fois une nouvelle fraction. La fonction `simplify` effectue son calcul directement dans la fraction passée en paramètre.



04 – Stylo

Mettre dans le fichier : pencil.h, src/*.c

Définissez dans votre entête le type suivant, puis implémentez la fonction indiquée.

```
typedef int (*t_pencil) (char c);  
int std_cputs(const char *s,  
              t_pencil pencil);
```

Le type `t_pencil` est un pointeur sur fonction. L'objectif de ce pointeur sur fonction est de déterminer la manière dont sera écrit le caractère. La fonction `tc_putchar` est compatible avec `t_pencil`.

La fonction `std_cputs` aura donc le même comportement que `std_puts` si utilisé avec `tc_putchar`.

Programmez en plus les fonctions `std_upchar`, `std_lowchar`, `std_revchar` qui afficheront le caractère en majuscule, en minuscule, et en casse inverse (minuscule si majuscule, majuscule si minuscule) et qui seront compatibles avec le type `t_pencil`.



05 – Votre bibliothèque

Créer à la racine de votre dossier personnel le dossier **libstd/**. Dans ce dossier **libstd/** vous disposerez un autre dossier **include/** ainsi que plusieurs autres dossiers. Votre politique de dossier respectera la table des normes, c'est à dire : un nom de dossier doit se traduire en préfixe de fonction, sauf si le nom du dossier est **src/**. Un préfixe doit se traduire en fichier en-tête.

Par exemple, la fonction **std_atoi** est dans **src/atoi.c**. La fonction **std_fraction_add** est dans **fraction/add.c** et le fichier **fraction.h** dans **include/**.

Dans votre dossier **libstd/** se trouvera un fichier **Makefile** qui compilera votre projet. Vous aurez également un dossier **tests/** qui contiendra un **Makefile** qui lui est propre. Ce dossier tests contiendra un dossier **src/** qui contiendra autant de fichier **.c** à compiler séparément que nécessaire. Ces fichiers **.c** contiendront les tests de votre bibliothèque.

Cette bibliothèque sera utilisé automatiquement par notre robot de correction lorsqu'il corrigera vos futurs projets. C'est à dire, que cette bibliothèque sera **copiée** dans l'environnement de correction. Faites bien attention à sa norme et à son fonctionnement car **tous vos travaux en pâtiront en cas de défaillance**.

La table est normes implique entre autre la présence de certaines règles dans le **Makefile**, ainsi que la présence d'une unique fonction **non static** dans chaque fichier **.c**. Votre bibliothèque ayant pour vocation d'être distribué en un seul bloc, chaque fonction non static doit être **unique**.

Vous trouverez un fichier archive **.tar.gz** (Tape archived gunzip) que vous pourrez décompresser avec la commande « **tar xvfz fichier.tar.gz** » contenant un modèle de projet complet, disposant des **Makefiles** correctement établis. Vous pouvez utiliser ces modèles pour vos projets, néanmoins, en examen, *ils ne vous seront pas disponible...*

Rangez dans votre dossier de bibliothèque l'ensemble des fonctions utilitaires programmées jusqu'ici, ainsi que les macros, c'est à dire :

repeat_char	to_digit	is_digit	strcasecmp	memmove	strcspn	merge
abs	pow	strlen	strncasecmp	memmem	strpbrk	reverse_split
is_printable	get_numeral	puts	strchrnul	memdup	strncat	cputs
is_between	print_base10	rputs	bzero	memcmp	ABS	new_fraction
clamp	print_base2	strcmp	atoi	swap	MIN	fraction_add
xor	from_digit	strchr	strcar	trim	MAX	fraction_mul
rot_minuschar	is_lower	strchr	strdup	strspn	CLAMP	fraction_sub
rclamp	is_upper	strcpy	strndup	strtol	RCLAMP	fraction_div
distance	to_lower	memset	strstr	print_base	DIFF	fraction_simplify
is_upper	to_upper	strlen	memcpy	sqrt	ARRAY_LENGTH	fraction_compute
is_lower	is_space	strncpy	substr	powf	print_split	
sum	is_blank	strncpy	strnstr	strtod	split	
factoriel	is_alpha	strncmp	strncasestr	strlcat	free_split	



Vous n'avez pas à fournir **toutes** ces fonctions avant la fin de la journée : faites de votre mieux. La correction automatique passera régulièrement sur votre bibliothèque et vous fournira les médailles des éléments que vous aurez pu y ajouter.

Par la suite, de nouvelles fonctions vous seront demandées. Il est attendu que toutes les fonctions utilitaires se retrouvent dans votre bibliothèque.

Certaines fonctions du système passeront pour vous d'interdites à autorisés sous conditions de réussites de certaines fonctions.