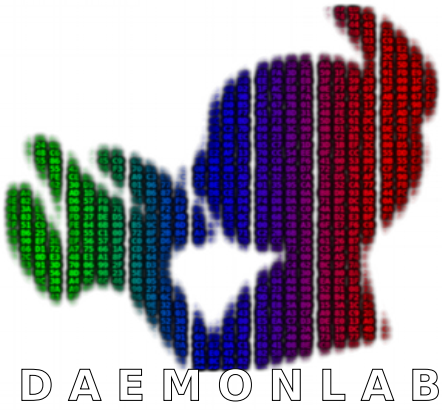

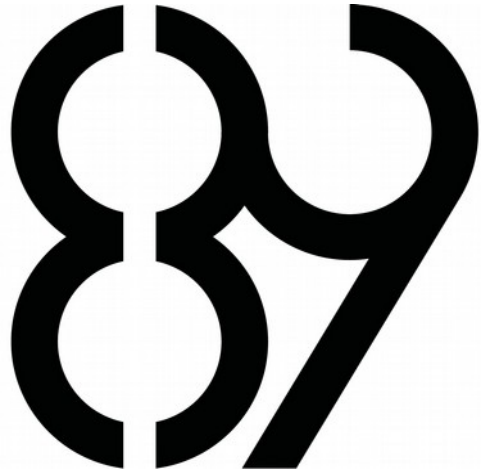

EXAM A/A

Examen de difficulté A

- DaemonLab -
infosphere@ecole-89.com

Médailles accessibles :

Définition des médailles à venir

INDEX

Avant-propos :

01 – Détails administratifs

Votre travail doit être envoyé via l'interface de ramassage du **TechnoCentre** :

http://technocentre.ecole89.com/ramassage

Le numéro de code présent sur ce sujet vous est propre : vous devrez le renseigner en rendant votre travail. En cas d'erreur, votre travail ne sera pas associée à l'activité et votre travail ne sera pas ramassé.

Pour cette activité, vous rendrez votre travail sous la forme d'une archive au format tar.gz. Cette archive devra contenir l'ensemble de votre travail tel que demandé dans la section 4.

Pour créer cette archive .tar.gz, il vous suffit d'utilliser la commande suivante :

```
$> tar cvfz mon_fichier.tar.gz fichier1 fichier2 fichier3
```

Le nom « mon_fichier.tar.gz » étant à remplacé par le nom que vous souhaitez donner votre fichier archive, et « fichier1 », « fichier2 », « fichier3 » par les fichiers ou dossiers que vous souhaitez mettre dans cette archive. Vous pouvez vérifier le contenu de votre archive à l'aide de la commande « **tar -t mon_archive.tar.gz** ».

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.

Médailles accessibles :

Réussir à rendre :
Vous avez réussi à envoyer votre travail au système de correction.

02 – Propreté de votre rendu

Votre rendu, c'est à dire le contenu de l'archive ou du dépôt que vous entrez sur l'interface du TechnoCentre, doit respecter **strictement** l'ensemble des règles suivantes :

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

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

Médailles accessibles :

Rendu propre
Votre rendu respecte les règles de propretés imposées.

03 – Règlement quant à la rédaction du code C

Vous devez respecter les règles des Tables de la Norme, sans exception.

04 – Construction de votre rendu

Le programme de correction va construire une sous-partie déterminée de votre rendu afin d'effectuer des tests dessus. En voici les paramètres :

- Les fichiers qui seront compilés sont ceux qui auront l'extension *.c.
- Seuls les fichiers dans le(s) dossier(s) ./ et sous-dossiers seront compilés.
- Les fichiers seront compilés avec -W -Wall -Werror.

- Tous les fichiers seront compilés **ensemble**, cela signifie donc que chaque fonction doit être unique, et que vous pouvez utiliser les fonctions des autres exercices dans chaque exercice.

L'ensemble du code que vous rendez doit pouvoir être compilé.
En cas d'échec de la compilation, vous ne serez pas évalué.

Médailles accessibles :

Construction partielle
Les éléments requis de votre projet se construisent séparément.

05 – Fonctions interdites

Vous n'avez le droit à aucune autre fonction que celle précisée dans la liste ci-dessous :

- putchar

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.

06 – Saut de ligne

Écrire un programme qui effectue un saut de ligne. Votre programme devra renvoyer 0.

07 – Distance

Implémentez la fonction suivante :

```
int                 e89_distance(int        a,
                                 int        b)
```

Cette fonction renvoi la distance entre a et b. Cette distance est toujours positive.

08 – Dix fois

Écrivez un programme qui affiche 10 fois le caractère ‘z’.

Votre fonction **main** ne devra pas dépasser 9 lignes.

Votre programme devra retourner 50.

09 – Letter

Implémentez la fonction suivante :

```
int             e89_display_letter(int        nbr)
```

Cette fonction affiche les lettres en minuscules si nbr est compris entre 0 et 25, en partant de 'a'. Si nbr est compris entre 26 et 51, il affichera les lettres en majuscule.

Si la fonction a affiché un caractère, elle renverra 1, sinon elle renverra 0.

10 – Carré

Implémentez la fonction suivante :

```
void            e89_print_square(int            width,
                                 int            height,
                                 char           letter)
```

Cette fonction affiche un carré de width de largeur et de height de hauteur, constitué de caractère letter. Ci-dessous, un exemple de 4 de largeur et de 6 de haut avec # comme caractère :

```
####
####
####
####
####
####
```

11 – Triangle

Implémentez la fonction suivante :

```
void            e89_print_triangle(int          size,
                                   char         inside,
                                   char         border)
```

Cette fonction affiche un triangle rectangle dont les deux cotés n'étant pas l'hypoténuse mesure **size**. Le caractère **inside** sera utilisé pour remplir l'intérieur, tandis que le caractère **border** sera utilisé pour l'extérieur. Ci dessous, un exemple de triangle, tel qu'il doit être affiché par votre programme, de taille 8, avec **inside** étant - et border # :

```
#
##
#-#
#--#
#---#
#----#
#-----#
########
```