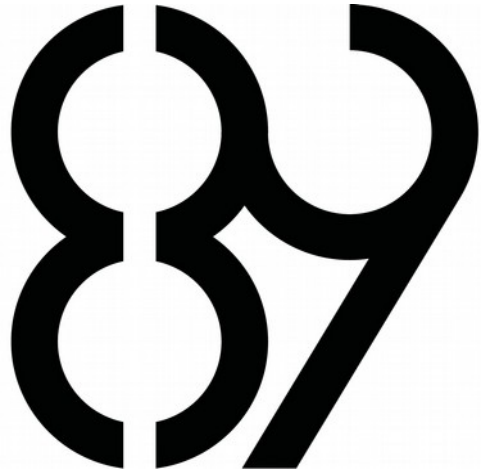
DAEMONLAB



TREE

Programmez la commande tree.
Faites pousser un bel arbre.

- DaemonLab -
pedagogie@ecole-89.com

INDEX

Avant-propos :

01 – Détails administratifs

Votre travail doit être envoyé via l'interface de ramassage de l'Infosphère :

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

Votre programme doit respecter la **Table des Normes**.

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) ./ 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.

Le nom de votre fichier exécutable sera etree.

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 :

- opendir
- readdir
- closedir

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 – Projet

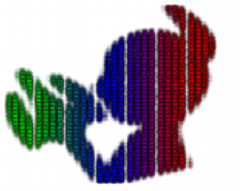
Votre version du logiciel tree affichera le contenu du dossier courant. Les dossiers situés dans le dossier courant seront parcouru **récursivement**. Le contenu de ces sous-dossiers sera précédé par une **indentation de deux espaces.** Les dossiers seront suivi du caractère '/'

Voici un exemple d'utilisation :

```
$> ls
abc def/ ghi jkl/
$> ls def/
mno pqr
$> ls jkl/
stu vwx/ yzA
$> ls jkl/vwx/
BCD EFG
$> tree
abc
def/
  mno
  pqr
ghi
jkl/
  stu
  vwx/
    BCD
    EFG
  yzA
```

Sur la page suivante, vous trouverez un exemple de code exploitant les fonctions dont vous avez besoin pour réaliser ce travail.

Vous aurez bien sur besoin de lire les manuels des fonctions **opendir**, **readdir** et **closedir** pour savoir tous ce que vous avez besoin de savoir.




```c
#include <dirent.h>

int main(void)
{
     DIR *dir ;
     struct dirent *browse;

     if ((dir = opendir("./")) == NULL)
          return (EXIT_FAILURE);
     while ((browse = readdir(dir)) != NULL)
     {
          puts(browse->d_name);
     }
     closedir(dir);
}
```