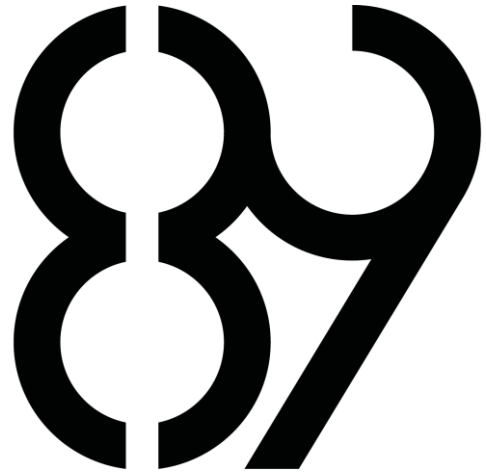




DAEMONLAB



Editor

Editeur de texte en ligne de commande

- DaemonLab -
infosphere@ecole-89.com

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



INDEX

Avant-propos :

- 01 – Détails administratifs
- 02 – Propreté de votre rendu
- 03 – Règlement quant à la rédaction du code C
- 04 – Construction de votre rendu
- 05 – Fonctions interdites
- 06 – Projet
- 07 – Informations complémentaires
- 08 – Pistes de recherches



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'utiliser la commande suivante : Le nom « mon_fichier.tar.gz » étant à remplacer par le nom que vous souhaitez donner à 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 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.

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** 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.**



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.**



05 – Fonctions interdites

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

- open
- close
- write
- read
- malloc
- free
- lseek
- fseek
- stat

**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

Vous devez développer un programme qui permet l'édition de fichier dans un terminal. Votre programme doit s'appeler « editor » et suivre les fonctionnalités suivantes.

Le programme peut se lancer avec un nom de fichier en paramètre ou bien se lancer seul :

```
cheap@DESKTOP-V1DPBII:~$ editor
```

```
cheap@DESKTOP-V1DPBII:~$ editor toto.txt  
10
```

Dans le cas où editor est lancé avec un paramètre, le programme affiche la taille du fichier. Dans les deux cas, votre programme va ensuite lire l'entrée utilisateur.

Pour commencer l'écriture d'un fichier :

- Vous devez écrire un « a » suivi d'un retour à la ligne ;
- Ecrire le contenu du fichier ;
- Finir par un « . » ;
- Ecrire « w » ;
- Afficher la taille du fichier avec le nouveau contenu.

La lettre « q » permet de quitter le programme.

```
cheap@DESKTOP-V1DPBII:~$ echo "tata" > toto.txt  
cheap@DESKTOP-V1DPBII:~$ cat toto.txt  
tata  
cheap@DESKTOP-V1DPBII:~$ editor toto.txt  
5  
a  
Hello I'm writing in toto.txt  
I can put multiples lines  
.  
w  
61  
q
```

```
cheap@DESKTOP-V1DPBII:~$ cat toto.txt  
tata  
Hello I'm writing in toto.txt  
I can put multiples lines
```



Par défaut, le curseur de l'éditeur se situe à la toute fin du fichier. Il est possible de connaître à quelle ligne le curseur se situe en utilisant la commande « n » ; la commande affiche le numéro de ligne suivi du contenu de la ligne :

```
cheap@DESKTOP-V1DPBII:~$ cat -n seven_taylor_swift
 1 Please picture me
 2 In the trees
 3 I hit my peak at seven feet
 4 In the swing
 5 Over the creek
 6 I was too scared to jump in
 7 But I, I was high in the sky
 8 With Pennsylvania under me
 9 Are there still beautiful things?
10 Sweet tea in the summer
11 Cross your heart, won't tell no other
12 And though I can't recall your face
13 I still got love for you
14 Your braids like a pattern
15 Love you to the moon and to Saturn
16 Passed down like folk songs
17 The love lasts so long
18 And I've been meaning to tell you
19 I think your house is haunted
20 Your dad is always mad and that must be why
21 And I think you should come live with
22 Me and we can be pirates
23 Then you won't have to cry
24 Or hide in the closet
25 And just like a folk song
26 Our love will be passed on
27 Please picture me
28 In the weeds
29 Before I learned civility
30 I used to scream ferociously
31 Any time I wanted
32 I, I
33 Sweet tea in the summer
34 Cross my heart, won't tell no other
35 And though I can't recall your face
36 I still got love for you
37 Pack your dolls and a sweater
38 We'll move to India forever
39 Passed down like folk songs
40 Our love lasts so long
```

```
cheap@DESKTOP-V1DPBII:~$ editor seven_taylor_swift
1053
n
40      Our love lasts so long
```




Il est possible de déplacer le curseur de l’éditeur en écrivant le numéro de ligne :

```
cheap@DESKTOP-V1DPBII:~$ editor seven_taylor_swift
1053
n
40      Our love lasts so long
4
In the swing
7
But I, I was high in the sky
8
With Pennsylvania under me
3
I hit my peak at seven feet
```

Liste des commandes à implémenter

a	Commence l’écriture d’un fichier
.	Arrête la sauvegarde de l’entrée standard
w [nom du fichier]	Ecris le contenu de la buffer dans le fichier ; peut être utilisé avec un argument ou sans
n	Affiche le numéro de ligne actuel suivi du contenu de la ligne actuelle
Un Nombre +/- (1-99999999999)	Déplace le curseur d’editor à la ligne correspondante



07 – Informations complémentaires

En cas d'erreur d'utilisation, vous devez afficher un message explicite pour que l'utilisateur comprenne son erreur.

Vous pouvez utiliser le programme « ed » comme référence d'utilisation pour « editor ».

Lorsque l'utilisateur écrit le contenu d'un fichier, l'ensemble du contenu est stocké dans une buffer ; donc pensez à malloc.

08 – Pistes de recherches

1. L'utilisateur peut rentrer un nombre infini de lignes ; par conséquent il faut que votre buffer grandisse au fur et à mesure que votre programme enregistre les entrées utilisateur. Une solution consiste à créer une liste chaînée qui ajoute les entrées utilisateurs au fur dans la liste. De la même manière il est possible de le faire en utilisant des tableaux mais cela implique une gestion habile de la mémoire de votre programme.
2. Pensez à placer chaque ligne du fichier lu sous la forme d'un tableau.