



Piscine - C - Tek1

Sujet Jour 04

Responsables Astek astek_resp@epitech.eu



Table des matières

| | |
|-----------|---|
| Consignes | 2 |
| Exo 1 | 3 |
| Exo 2 | 4 |
| Exo 3 | 5 |
| Exo 3 bis | 6 |
| Exo 4 | 7 |
| Exo 5 | 8 |



Consignes

- Le sujet peut changer jusqu'à une heure avant le rendu.
- Vos exercices doivent être à la norme.
- Vous ne devez avoir de `main()` dans aucun fichier de votre repertoire de rendu.
- Pour chaque repertoire de chaque exercice nous allons compiler vos fichiers avec la commande `cc -c *.c`, ce qui va générer tous les fichiers `.o` que nous allons ensuite linker un par un en y ajoutant notre `main.c` et notre `my_putchar.c` :

```
$> cd ex_01
$> cc -c *.c
$> cc *.o ~moulinette/main_ex_01.o ~moulinette/my_putchar.o -o ex01
$> ./ex01
[...]
```

- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
Si un seul de vos fichiers empêche la compilation avec `*.c`, la moulinette ne pourra pas vous corriger et vous aurez 0. Vous avez donc tout intérêt à effacer vos rendus d'exercices ne fonctionnant pas.
- Vous n'avez le droit qu'à la fonction `my_putchar` pour faire les exercices qui suivent. Cette fonction sera fournie, donc :
 - vous ne devez pas avoir lors du rendu de fichier `my_putchar.c`
 - la fonction `my_putchar` ne doit être mise dans aucun des fichiers rendus
- Pensez à en discuter sur le forum piscine !
- Travaillez en local !
C'est-à-dire que pour chaque exercice vous devez le compiler sur votre compte linux puis, une fois qu'il fonctionne, le copier sur votre compte AFS.
Ceci dans le simple but de ne pas surcharger les serveurs car vous êtes nombreux.



Indices Faites vous un script shell pour copier vos fichiers sur l'AFS



Exo 1

- Écrire une fonction qui échange le contenu de deux pointeurs sur entiers.
- Ces deux pointeurs sont les paramètres de cette fonction.
- Elle devra être prototypée de la façon suivante :

```
1  int my_swap(int *a, int *b);
```

- Rendu :
/afs/epitech.net/users/group/login/rendu/piscine/Jour_04/ex_01/my_swap.c



Exo 2

- Écrire une fonction qui affiche un à un les caractères d'une chaîne à l'écran.
- L'adresse du premier caractère de la chaîne est contenue dans le pointeur passé en paramètre à la fonction.
- Elle devra être prototypée de la façon suivante :

```
1 int my_putstr(char *str);
```

- Rendu :
/afs/epitech.net/users/group/login/rendu/piscine/Jour_04/ex_02/my_putstr.c



Exo 3

- Écrire une fonction qui compte le nombre de caractères dans une chaîne de caractères et qui retourne le nombre trouvé.
- Elle devra être prototypée de la façon suivante :

```
1 int my_strlen(char *str);
```

- Rendu :
/afs/epitech.net/users/group/login/rendu/piscine/Jour_04/ex_03/my_strlen.c



Exo 3 bis

- Le but de cet exercice est d'inverser chaque caractère de la chaîne 'str' deux à deux. Ainsi, on inversera le premier avec le dernier, le deuxième avec l'avant-dernier, et ainsi de suite.
- Cette fonction renverra un pointeur sur le premier caractère de la chaîne nouvellement inversée.

```
1 char *my_evil_str(char *str);
```

- Rendu :
/afs/epitech.net/users/group/login/rendu/piscine/Jour_04/ex_03b/my_evil_str.c
- Exemple :

```
1 a => a
2 ab => ba
3 abc => cba
4 abcd => dcba
5 abcde => edcba
6 abcdef => fedcba
7 ...
```



Exo 4

- Écrire une fonction qui renvoie un nombre. Ce nombre est connu sous la forme d'une chaîne de caractères passée en paramètre (Ex : "842") et doit être transformé en int.
- On considèrera que ce nombre est exprimé en base 10 ("0123456789").
- La fonction devra être prototypée de la façon suivante :

```
1  int my_getnbr(char *str);
```

- La fonction doit gérer les chaînes suivantes :
 - "+-----+-----+-----+-----42" retourne le nombre -42
 - "42a43" retourne le nombre 42
 - "1100000000000000000000000000000042" retourne 0 car ce nombre ne peut pas être contenu dans un int.
 - "-1000000000000000000000000000000042" retourne 0 car ce nombre ne peut pas être contenu dans un int.
- Rendu :
/afs/epitech.net/users/group/login/rendu/piscine/Jour_04/ex_04/my_getnbr.c



Exo 5

- Écrire une fonction qui trie un tableau d'entiers, par ordre croissant.
- Les paramètres sont : un pointeur sur entier, le nombre d'entiers dans le tableau.

```
1 void my_sort_int_tab(int *tab, int size);
```

- Rendu :
/afs/epitech.net/users/group/login/rendu/piscine/Jour_04/ex_05/my_sort_int_tab.c