



Piscine - C - Tek1

Sujet Jour 08

Responsable Astek [astek\\_resp@epitech.eu](mailto:astek_resp@epitech.eu)



## Table des matières

Consignes	2
Exercice 1 - my_strdup	3
Exercice 2 - convert_base	4
Exercice 3 - sum_params	5
Exercice 4 - my_str_to_wordtab	6
Exercice 5 - my_show_wordtab	7



# 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.
- 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

- Nous ramassons tous les `.c` du répertoire et nous compilerons avec votre `libmy` se trouvant dans `/afs/epitech.net/users/group/login/rendu/lib/my`.
- Pour ceux qui utilisent des `.h`, il doivent se trouver dans :  
`/afs/epitech.net/users/group/login/rendu/include`
- Toutes les fonctions nécessaires à la compilation de l'exercice doivent se trouver soit dans le `.c` spécifié par le sujet soit dans la `lib`.
- La fonction demandée dans le sujet doit être dans le `.c` demandé.
- Si par exemple, vous utilisez `my_strdup` pour `my_str_to_wordtab` (`my_strdup` devra se trouver dans le `.c` de `my_str_to_wordtab` car il ne peut être dans la `lib` car demandé à l'exo 1.)
- Rendu :  
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_08`



Attention aux droits



## Exercice 1 - my\_strdup

- Écrire une fonction qui alloue suffisamment de mémoire et copie la chaîne de caractères passée en argument.
- Elle devra être prototypée de la façon suivante :

```
1 char *my_strdup(char *src);
```

- Elle doit renvoyer un pointeur sur la nouvelle chaîne allouée.
- Rendu :  
[/afs/epitech.net/users/group/login/rendu/piscine/Jour\\_08/ex\\_01/my\\_strdup.c](/afs/epitech.net/users/group/login/rendu/piscine/Jour_08/ex_01/my_strdup.c)



*Indices* Indice : man strdup



## Exercice 2 - convert\_base

- Écrire une fonction qui renvoie le résultat de la conversion de la chaîne **nbr** exprimée en une base **base\_from** dans une base **base\_to** sous forme d'une chaîne de caractères allouée avec suffisamment de mémoire. Le nombre représenté par **nbr** tient dans un **int**.
- Elle devra être prototypée de la façon suivante :

```
1 char *convert_base(char *nbr, char *base_from, char *base_to);
```

- Rendu :  
[/afs/epitech.net/users/group/login/rendu/piscine/Jour\\_08/ex\\_02/convert\\_base.c](/afs/epitech.net/users/group/login/rendu/piscine/Jour_08/ex_02/convert_base.c)



## Exercice 3 - sum\_params

- Écrire une fonction qui tranforme les arguments reçus en ligne de commande en une unique chaîne de caractères. Les arguments seront séparés par un '\n'.

- Exemple :

```
1  int main(int ac, char **av)
2  {
3      my_putstr(sum_params(ac, av));
4      return (EXIT_SUCCESS);
5  }
```

```
1  (foo_b@fedora) cc -o sum_params sum_params.c main.c my_putstr.c
2  (foo_b@fedora) ./sum_params toto titi | cat -e
3  ./sum_params$
4  toto$
5  titi(foo_b@fedora)
```

- Elle devra être prototypée de la façon suivante :

```
1  char *sum_params(int argc, char **argv);
```

- Rendu :

[/afs/epitech.net/users/group/login/rendu/piscine/Jour\\_08/ex\\_03/sum\\_params.c](/afs/epitech.net/users/group/login/rendu/piscine/Jour_08/ex_03/sum_params.c)



## Exercice 4 - my\_str\_to\_wordtab

- Écrire une fonction qui découpe une chaîne de caractères en mots.
- Les séparateurs sont tous les caractères non-alphanumériques.
- La fonction renvoie un tableau, où chaque case contient l'adresse d'une chaîne de caractères représentant un mot. Le dernier élément du tableau devra être égal à 0 pour marquer la fin du tableau.
- La chaîne qui sera transmise sera modifiable et modifiée dans votre fonction.
- Elle devra être prototypée de la façon suivante :

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

- Rendu :

[/afs/epitech.net/users/group/login/rendu/piscine/Jour\\_08/ex\\_04/my\\_str\\_to\\_wordtab](/afs/epitech.net/users/group/login/rendu/piscine/Jour_08/ex_04/my_str_to_wordtab)



## Exercice 5 - my\_show\_wordtab

- Écrire une fonction qui affiche le contenu du tableau créé dans l'exercice précédent.
- Chaque mot sera seul sur une ligne.
- Chaque mot sera suivi d'un `\n`, y compris le dernier.
- Cet exercice sera compilé avec votre `my_str_to_wordtab.c`
- Faites attention à ne pas faire de multiples `define`.
- Elle devra être prototypée de la façon suivante :

```
1 int my_show_wordtab(char **tab);
```

- Rendu :

`/afs/epitech.net/users/group/login/rendu/piscine/Jour_08/ex_05/my_show_wordtab.c`