Piscine - C - Tek1

Sujet Jour 03

Responsables Astek astek_resp@epitech.eu

Table des matières

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 Pour chaque repertoire de chaque exercice nous allons compiler vos fichiers avec la commande `cc -c *.c`, ce qui va génerer 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 specifiés par les énoncés des exercices.
  Si un seul de vos fichiers empèche la compilation avec `*.c`, la moulinette ne pourra pas vous corriger et vous aurez `0`. Vous avez donc tout intéret 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`

- Pour tous les exercises les tableaux sont interdits !
- Dossier de rendu :
  `/afs/epitech.net/users/group/login/rendu/piscine/Jour_03`

Exo 01 : my_aff_alpha

- Écrire une fonction qui affiche l'alphabet en minuscule sur une seule ligne, dans l'ordre croissant à partir de la lettre 'a'.
- Elle devra être prototypée de la façon suivante :

```
1   int my_aff_alpha();
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_03/my_aff_alpha.c`

Exo 02 : my_aff_revalpha

- Écrire une fonction qui affiche l'alphabet en minuscule sur une seule ligne, dans l'ordre décroissant à partir de la lettre 'z'.
- Elle devra être prototypée de la façon suivante :

```
1   int my_aff_revalpha();
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_03/my_aff_revalpha.c`

Exo 03 : my_aff_chiffre

- Écrire une fonction qui affiche tous les chiffres sur une seule ligne dans l'ordre croissant.
- Elle devra être prototypée de la façon suivante :

```
1   int my_aff_chiffre();
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_03/my_aff_chiffre.c`

Exo 04 : my_isneg

- Écrire une fonction qui affiche 'N' ou 'P' suivant le signe de l'entier passé en paramètre. Si n est négatif alors afficher 'N'. Si n est positif ou nul alors afficher 'P'.
- Elle devra être prototypée de la façon suivante :

```
1   int my_isneg(int n);
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_03/my_isneg.c`

Exo 05 : my_aff_comb

- Écrire une fonction qui affiche dans l'ordre croissant toutes les différentes combinaisons de trois chiffres différents dans l'ordre croissant.
- Cela donne quelque chose comme ça :
"012, 013, 014, 015, 016, 017, 018, 019, 023, ..., 789"
- 987 n'est pas là car 789 est déjà présent
- 999 n'est pas là car ce nombre ne comporte pas des chiffres exclusivement différents les uns des autres
- Elle devra être prototypée de la façon suivante :

```
1   int my_aff_comb();
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_03/my_aff_comb.c`

Exo 06 : my_aff_comb2

- Écrire une fonction qui affiche toutes les différentes combinaisons de deux nombres entre 0 et 99, dans l'ordre croissant.
- Cela donne quelque chose comme ça :
"00 01, 00 02, 00 03, 00 04, 00 05, ..., 01 99, 02 03, ..., 98 99"
- Elle devra être prototypée de la façon suivante :

```
1   int my_aff_comb2();
```

- Rendu :
`/afs/epitech.net/users/group/login/rendu/piscine/Jour_03/my_aff_comb2.c`

Exo 07 : my_put_nbr

- Écrire une fonction qui affiche un nombre passé en paramètre. La fonction devra être capable d'afficher la totalité des valeurs possibles dans une variable de type `int`.
- Elle devra être prototypée de la façon suivante :

```
1    void my_put_nbr(int nb);
```

- Par exemple :
  - my_put_nbr(42) affiche "42"
  - my_put_nbr(0) affiche "0"
  - my_put_nbr(-42) affiche "-42"
  - my_put_nbr(2147483647) affiche "2147483647"
  - my_put_nbr(-2147483648) affiche "-2147483648"
- Rendu :
  `/afs/epitech.net/users/group/login/rendu/piscine/Jour_03/my_put_nbr.c`

Exo 08 : my_aff_combn

- Écrire une fonction qui affiche toutes les différentes combinaisons de n chiffre(s), dans l'ordre croissant.
- Si n = 2, cela donne quelque chose comme ça :
  "01, 02, 03, ..., 09, 12, ..., 79, 89"
- Elle devra être prototypée de la façon suivante :

```
1   int my_aff_combn(int n);
```

- Rendu :
  `/afs/epitech.net/users/group/login/rendu/piscine/Jour_03/my_aff_combn.c`