

TP2

Les fichiers doivent être rendu dans votre github le nom du projet est [pt_tp2.git](https://github.com/pt-tp2)

Respecter la norme de codage.

Pas de warning avec l'option de compilation `-Wall`

Vous **devez pousser** vos fichiers en fin de TP, vous pouvez faire un deuxième commit/push le soir du TP avant minuit pour récupérer des points supplémentaires.

Utilisez l'option `-o` de gcc pour nommer votre exécutable (il ne s'appellera plus `a.out`)

Ce TP n'est pas noté

Exercice 0

Ecrire une fonction qui prend en paramètre une chaîne de caractère et retourne celle-ci en majuscule.

Notes:

- Traiter les cas d'erreur éventuels.
- La fonction doit être mise dans un fichier nommé `tp2.c`
- Il ne doit pas y avoir de fonction `main` dans le fichier `tp2.c`
- Le fichier `tp2.h` contient le prototype de la fonction.
- Ne pas utiliser des fonctions existantes.
- Vous devez écrire un `main` dans un autre fichier pour tester (exemple: `ex0.c`)

Voici le prototype de la fonction:

```
char* upper(char *s)
```

Exercice 1

Ecrire un programme qui prend en paramètre une chaîne de caractère et l'affiche en forme de pyramide

Important:

- Traiter les cas d'erreur éventuels.
- Vous pouvez utiliser le code du TP1

exemple:

```
$ ./ex1 "tota"
atotota
 totot
  oto
   t
```

Exercice 2

Ecrire un programme qui prend en paramètre une chaîne de caractères et l'affiche fractionnée comme l'exemple. Le premier caractère de chaque ligne est en majuscule.

Notes:

- Traiter les cas d'erreur éventuels.
- Le programme est écrit dans un fichier nommé ex2.c
- Vous ne pouvez pas utiliser strtok
- Attention au majuscule.

Exemple d'exécution:

```
$ ./ex2 " Salut ceci est un test "  
Salut  
Ceci  
Est  
Un  
Test
```

Exercice 3

Ecrire un programme qui prend en paramètre une chaîne de caractères représentant un entier positif et qui le retourne sous forme d'entier.

- Traiter les cas d'erreur éventuels.
- Le programme est écrit dans un fichier nommé ex3.c
- Vous ne pouvez pas utiliser la fonction atoi

La fonction main ressemblera à ça:

```
int main(.....) {  
    int ret_val = 0;  
  
    ret_val = convert(.....);  
    return (ret_val);  
  
}
```

Exemple:

```
$ ./ex4 "12345"
```

Pour vérifier le résultat taper la commande

```
echo $?
```