

PANDALAB

BITMEL

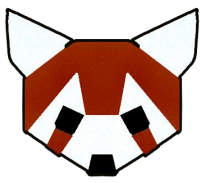
L'ENFER DES BITS !

pedagogie@ecole-89.com

Attention à ne pas emmêler le fil !

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

Auteur : Elie



Avant-Propos :

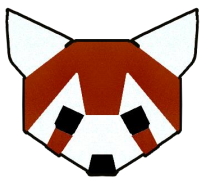
01 – Avant-Propos

02 – Fonctions interdites

03 – Médailles accessibles

Exercices :

04 - Sujet



01 – Avant-propos

Votre travail doit être rendu via le répertoire git 2022_BITMEL.

Si vous faites erreur et que le dossier que vous utilisez pour votre rendu est différent, vous ne serez pas évalué faute d'avoir pu trouver votre travail.

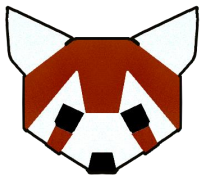
Votre rendu doit respecter strictement l'ensemble des règles suivantes :

- 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)**

Toute fonction « main » dans votre rendu mettra fin définitive à la correction !

Vos fichiers doivent dépendre de panda.h.

Votre programme doit respecter les Tables de la Norme dans leur intégralité. Vous êtes invité à les observer depuis l'Infosphère. Elles sont disponibles comme ressource de cette activité.



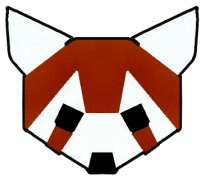
02 – Fonctions interdites

La bibliothèque logicielle venant avec le C est vaste et disponible. Nous avons cependant fait le choix de vous interdire son utilisation, afin de vous amener progressivement à reprogrammer vous-même ses fonctionnalités les plus utiles.

L'usage d'une fonction interdite est assimilé à de la triche. La triche provoque l'arrêt de l'évaluation et la perte des médailles.




Au cours de la journée, si vous estimez qu'une fonction interdite est nécessaires, vous pouvez demander à ce qu'elle soit autorisée, en vous justifiant.

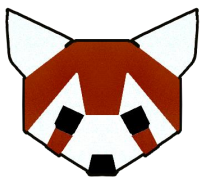
Vous n'avez le droit d'utiliser aucune fonction issue de la LibC à l'exception de celles que nous vous autoriserons explicitement.



03 – Médailles accessibles

Voici ci-dessous la liste des médailles accessibles via cette activité :

Dépôt propre		
Utilisation de git simple		
bitshift		
pointer		
binary_or		
binary_and		
binary_not		



GETBIT

Implémentez la fonction suivante qui a pour effet de retourner la valeur du bit contenu à la position *i* de l'espace mémoire *c*.

```
size_t std_getbit (char c, int i)
```

Set and Remove!

Implémentez les fonctions permettant de mettre à 1 et 0 les bits aux positions *i*.

```
void std_set_bit(char *test, int i);  
void std_remove_bit(char *test, int i);
```

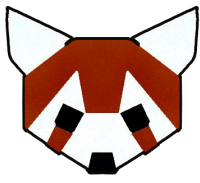
Do you know Mask ?

Implémentez la fonction suivante, qui a pour effet de renvoyer le paramètre avec les 4 premiers bits inversé avec les 4 derniers bits tels que :

10100001 → 00011010

Pour cet exercice nous vous invitons à utiliser les masques de bit !

```
char std_swap_bit(char test)
```



BIT CHAKE (Did you miss it?)

Implémentez la fonction suivante,

Cette fonction va échanger le bit défini par la valeurs des 3 premiers bits de la clef (par char) et l'échanger avec le char contenu à l'index défini par les 5 derniers bits de la clef.

```
void std_bitshake(char *cnt,  
                  size_t len,  
                  char *key,  
                  bool cipher)
```

KEY : « 123 »

SOURCE : « Pokemon »

RESULT : « Vkigmol »

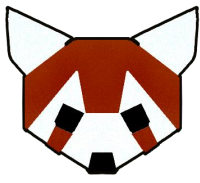
CIPHER : FALSE

KEY : « 123 »

SOURCE : « Pokemon »

RESULT : « Vkigmmn »

CIPHER : TRUE



ByBit (Did you understand it?)

Implémentez la fonction suivante,

Cet encryptage, va avoir pour effet de regrouper les bit en fonction de leurs valeurs

Turn Oribyt:

01234567 01234567 01234567 01234567 01234567 01234567 01234567 01234567

Into Tarbyt:

00000000 11111111 22222222 33333333 44444444 55555555 66666666 77777777

```
void std_bybit (char *cnt,  
               size_t len)
```

Switch (Turn this up!)

Implémentez la fonction suivante,

Cet encryptage va avoir pour effet de changer les bit tels qu'ils sont en effet de leurs changements d'état, un 1 implique un changement d'état un 0 que l'état perdure. Le premier bit indique le niveau initial !

1100 deviendras 1010

```
void std_switch (char *cnt,  
                size_t len,  
                char *key,  
                bool cipher)
```