

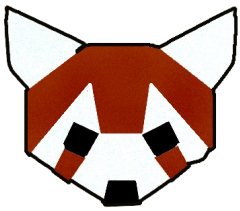
PANDALAB

EXAM B/C

Examen de difficulté B

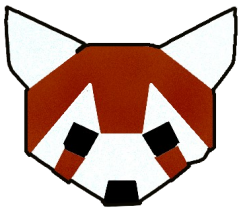
- PandaLab -

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



INDEX

- 01 – Avant-propos
- 02 – Fonctions autorisées
- 03 – minputs
- 04 – recputs
- 05 – trim
- 06 – strchr
- 07 – Bonus : print_base10



01 – Avant-propos

Votre travail doit être rendu via le dossier `~/examen/exam0/` dans votre espace personnel. Vous n'avez pas le droit à votre bibliothèque et vos travaux doivent être tous rendu dans ce dossier.

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.

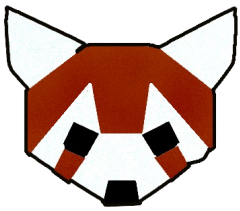
Ce travail est à effectuer seul. De plus il s'agit d'un examen. Vous n'avez donc pas le droit de communiquer avec vos camarades.

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)

La présence d'un fichier interdit mettra immédiatement fin à votre évaluation.

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 autorisées

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

Vous n'avez le droit d'utiliser aucune fonction issue de la LibC à l'exception de celles que nous vous autoriserons explicitement. Dans le cadre de cette activité, la seule fonction à laquelle vous aurez le droit est l'appel système **write**.

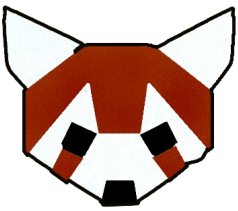
Vous pouvez observer le fonctionnement de **write** en tapant « **man 2 write** » dans votre terminal, néanmoins, afin de vous faciliter son utilisation au départ, nous vous avons préparé une fonction que vous êtes libre d'intégrer à votre code.

Cette fonction est **tc_putchar**, elle tente d'écrire le caractère passé en paramètre sur la sortie standard (le terminal, par défaut). Elle renvoi 1 si elle a réussie à écrire le caractère, sinon elle renvoi 0. Ci-dessous, le code de **tc_putchar** suivi d'un programme de test.

```
#include <unistd.h>

int tc_putchar(char c)
{
    return (write(1, &c, 1) > 0);
}
```

```
int main(void)
{
    tc_putchar('A');
    tc_putchar('\n');
    return (0);
}
```



03 – minputs minputs.c

Écrire un programme qui affiche ce qui est contenue dans **str** en transformant les majuscule en minuscule. Tout comme **puts**, la fonction effectue donc un saut de ligne à la fin.

Votre programme devra renvoyer le nombre de caractère écrits. Y compris le saut de ligne...

```
int std_minputs(const char *str)
```

04 – recputs recputs.c

Implémentez la fonction suivante :

```
int std_recputs(const char *str)
```

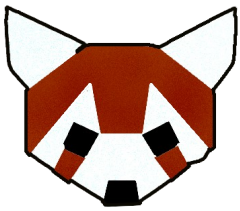
Cette fonction agit comme **puts**. Vous êtes cependant contraint à une construction spécifique : vous n'avez pas le droit, ni à **for**, ni à **while**, ni à **do while**, ni à **goto**. Vous ne pouvez pas faire de boucle, vous devez construire une fonction **récursive**.

05 – trim trim.c

Écrivez cette fonction qui nettoie les espaces de début et fin de chaînes de caractères.

```
void std_trim(char *str)
```

Notez que vous devez modifier directement les caractères contenu par **str** et non créer une nouvelle chaîne.



06 – strchr strchr.c

Implémentez la fonction suivante :

```
char *std_strchr(const char *str,  
char token)
```

Cette fonction est documentée par le système de manuel.

07 – Bonus : printbase10 printbase10.c

Implémentez la fonction suivante :

```
int std_printbase10(int nbr)
```

Cette fonction écrit le chiffre en base 10 sur le terminal et renvoi le nombre de caractères écrits.