

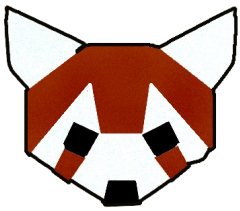
PANDALAB

EXAM PANDAPHIQUE

Examen de difficulté Panda

- PandaLab -

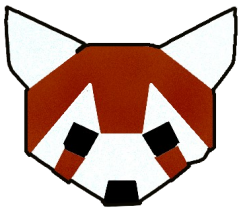
Ce document est strictement personnel et ne doit en aucun cas être diffusé.



INDEX

- 01 – Avant-propos
- 02 – Fonctions autorisées

- 03 – set_pixel
- 04 – Damier
- 05 – Carré
- 06 – Cercle
- 07 – Bonus : Pandelium



01 – Avant-propos

Votre travail doit être rendu via le dossier **~/hyperespace/exam3/** dans votre espace personnel.

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.

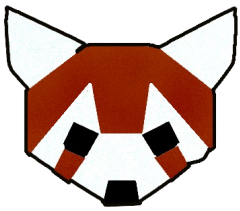
Ce travail est à effectuer seul. De plus il s'agit d'un examen. Vous n'avez donc pas le droit de communiquer avec vos camarades.

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.

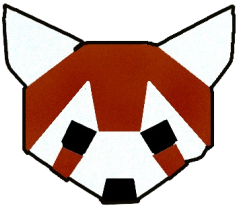


02 – Fonctions autorisées

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.



03 – set_pixel setpixel.c

Votre programme doit afficher un pixel de couleur « color » à la position pos de l'image px.

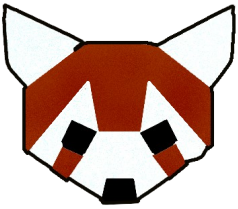
```
void      std_set_pixel(t_bunny_pixelarray *px,  
                      t_bunny_position pos,  
                      unsigned int color)
```

04 – Damier damier.c

Implémentez la fonction suivante :

```
void      std_damier(t_bunny_pixelarray *px,  
                   int size)
```

Cette fonction affiche un damier où les cases font size pixels de hauteur et de largeur.



05 – Carré square.c

Implémentez la fonction suivante :

```
void      std_set_square(t_bunny_pixelarray *px,  
                        t_bunny_position pos,  
                        int size)
```

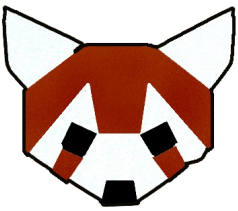
Cette fonction dessine un carré (vide) à la position `pos`, et faisant `size` longueur et hauteur.

06 – Cercle circle.c

Implémentez la fonction suivante :

```
void      std_set_circle(t_bunny_pixelarray *px,  
                        t_bunny_position pos,  
                        int radius)
```

Cette fonction dessine un cercle de centre `pos`, et de rayon `radius`.



07 – Bonus : Pandelium pandelium.c

Implémentez la fonction suivante :

```
void Pandelium (int nbr)
```

Cette fonction effectue la fractales suivante avec nbr profondeur.

