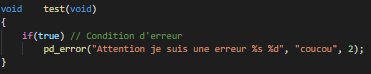
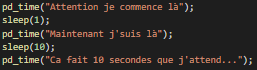
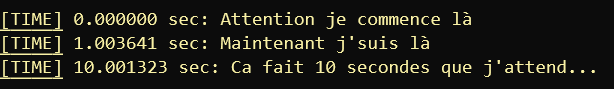
PARTIE 1 DAST (fonctions diverses)

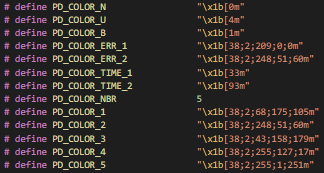
comment free ?  
pd\_free(x) == ft\_memdel((void\*\*)&(x))

Comment malloc ?  
pd\_malloc(sizeof(x) \* y) == renvoi un null si la taille est de 0 contrairement au malloc

Comment proteger un malloc pour return NULL ?  
  
Si le malloc return NULL, alors la macro va return NULL

Comment gerer une erreur ? (MACRO PRINTF, VA\_ARG)  
  
Cela va renvoyer dans la sortie d’erreur  
  
Puis ca va **exit(1)**

Comment calculer le temps d’éxecution entre deux etapes dans le code ? (MACRO PRINTF, VA\_ARG)  
  
Va donner cette sortie  


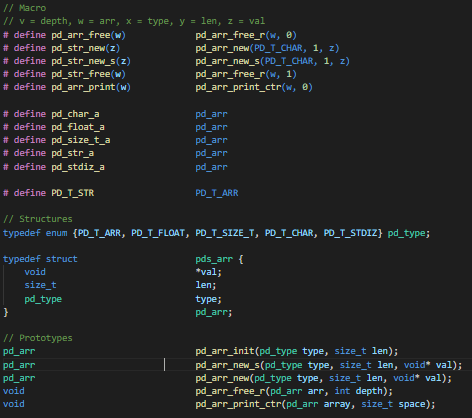
Comment utiliser les couleurs ?  
  
Si on veut utiliser plusieurs couleur par rapport a un nombre/index, utiliser la variable global  
pd\_color\_t avec pd\_color\_t[i % PD\_COLOR\_NBR]

PARTIE 2 ARRAY  
Comment créer un array ?

**pd\_arr** array = pd\_arr\_init(PD\_T\_FLOAT, 5) == creer un tableau de float de taille 5  
Mais quand on code on va remplacer le **pd\_arr** par **pd\_float\_a** (float array) pour plus de clareté meme en fait, **pd\_float\_a** == **pd\_arr** donc:  
**pd\_float\_a** = pd\_arr\_init(PD\_T\_FLOAT, 5)

Pour acceder a un element,  
if (array.type == PD\_T\_FLOAT)  
 ((float\*)array.val)[i]

Screen de pd\_arr.h



* un “s” à la fin d’une fonction comme **pd\_arr\_new\_s** signifie qu’on va prendre un tableau statique et le copier pour generer un **t\_arr**
* un “r” à la fin d’une fonction comme **pd\_arr\_free\_r** signifie qu’on va free en recursion donc si le tableau contient un tableau qui contient un tableau, soit on indique 2, soit on indique -1 qui va free tout les arrays. On observe que la macro **pd\_str\_free** utilise **pd\_arr\_free\_r** avec 1 de depth.

Que veut dire “ctr” à la fin d’une fonction ?  
Par exemple pour **pd\_arr\_print\_ctr(array, space),** cela veut dire que la fonction a une macro associé qui enleve le ctr donc **pd\_arr\_print(array) == pd\_arr\_print\_ctr(array, 0);** ca permet d’initialiser l’argument d’une recursive a 0