

Exercice 1 : Fonctions et tableaux

Ecrire la fonction `stddev(x, n)` qui retourne l'écart type des éléments du tableau `x` (réels) de dimension `n`. L'indilage se fait par rapport à 0. On dispose de la fonction `pow(a, b)` retournant a^b , qui sera utilisée uniquement pour le calcul de la racine, le carré étant obtenu par la multiplication.

Pour rappel, l'écart type est défini comme suit :

$$\sigma_X = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{\frac{1}{n} \left(\sum_{i=1}^n x_i^2 \right) - \bar{x}^2}$$

Exercice 2 : Chaînes

Ecrire l'algorithme de la procédure `align64(chaine)` permettant de réaliser un alignement à droite d'une chaîne de caractères sur 64 octets. Si la longueur de la chaîne est inférieure à 64, l'alignement à droite consistera à insérer autant d'espaces que nécessaire en début de chaîne de sorte que la longueur finale de la chaîne sera 64. Vous n'utiliserez aucune fonction annexe de type `strlen`, etc.

Exercice 3 : Calcul de durée

Ecrire l'algorithme de la procédure `duree(t1, t2)` calculant la durée entre `t1` et `t2` (`t2` postérieur à `t1`), `t1` et `t2` étant 2 temps distincts donnés sous forme de tableaux à 3 éléments entiers (h,m,s). Le résultat sera placé dans `t1`.

Exemple :

Si `t1=[18,40,50]` (18 heures, 40 minutes, 50 secondes)

Et `t2=[22, 10, 0]` (22 heures, 10 minutes, 0 secondes)

Alors on aura au final `t1=[3,29,10]` soit 3 heures, 29 minutes et 10 secondes.

Si `t2` est antérieur à `t1`, on considérera qu'il s'agit d'un temps du jour suivant. On ne dispose pas de fonction de comparaison de temps et l'argument `t2` ne doit pas être modifié après appel de la procédure `duree`.

Exercice 4 : Automatismes

Routine d'approche de la sonde d'un microscope à effet tunnel électronique.

Le microscope comporte une sonde conductrice placée à l'extrémité d'un vérin piézoélectrique vertical (axe Z). On réalise un balayage d'une surface durant lequel on contrôle le contact entre l'électrode et l'objet (également conducteur de l'électricité), tout en relevant l'altitude `z` de la sonde (partie à ne pas gérer dans ce problème). Si aucun courant ne circule, cela signifie qu'il n'y a pas de contact, que la sonde est donc trop éloignée et qu'il faut par conséquent s'approcher (descendre). Inversement si le courant est trop fort, il faut s'éloigner.

On dispose de :

- la fonction `Mesure()`, qui ne prend pas d'argument, et qui retourne un entier représentant la valeur du courant qui circule entre la sonde et l'échantillon s'il y a contact. Peu important les unités (en principe, des nano-Ampères).
- La fonction `Verin_Z(z)`, qui ne retourne rien, et qui admet comme paramètre un entier relatif indiquant l'incrément de déplacement relatif du vérin. Le vérin descend (approche la sonde de l'échantillon) pour un incrément positif ($z = 1$) ou monte (s'éloigne de l'échantillon) pour $z = -1$.

Ecrire la fonction `Palper()`, qui ne retourne rien et ne prend pas d'argument. Cette fonction gère la mise en contact de la sonde avec l'objet. Elle approche la sonde de l'objet par pas successifs de 1 si la mesure du courant est inférieure à 10, ou l'en éloigne si la mesure du courant est supérieure à 10, jusqu'à la l'obtention de la valeur 10.