INF1 - TD 5

UNIVERSITÉ DE RENNES 1

Exercice 1 :

1. Écrire une fonction COPIE qui prend en entrée un tableau d'entiers et qui en retourne une copie.
2. Écrire une fonction COMPARETAB qui prend comme paramètre deux tableaux d'entiers (`tab1` et `tab2`) qui possèdent la même longueur. La fonction doit renvoyer un tableau de caractères (**char**) de même longueur et qui contient les caractères $<$, $=$ et $>$. Le i-ème élément du tableau renvoyé contient $<$ si `tab1`$[i] <$ `tab2`$[i]$ (même chose pour $=$ et $>$).

Exercice 2 : Fusion

1. Écrire une fonction CONCATENATION qui prend en paramètres deux tableaux `tab1` et `tab2` de longueurs arbitraires et qui retourne un nouveau tableau qui contient les éléments de `tab1` puis les éléments de `tab2`.
2. Écrire une fonction FUSION qui prend comme paramètre deux tableaux d'entiers de longueurs quelconques. La fonction doit renvoyer un nouveau tableau d'entiers qui contient les éléments des deux tableaux reçus en paramètre intercalés. Par exemple, soit $s_1, s_2, \ldots, s_n$ les éléments du premier tableau et $t_1, t_2, \ldots, t_m$ les éléments du second tableau. Si $n \geq m$ alors les éléments du tableau renvoyé sont $s_1, t_1, s_2, t_2, s_3, t_3, \ldots, s_m, t_m, s_{m+1}, s_{m+2}, \ldots, s_n$.

Exercice 3 : Tableaux

1. Écrire une fonction NOMBREDEMULTIPLES qui prend en paramètre un tableau d'entiers `tab` ainsi qu'un entier n et qui retourne le nombre de multiples de n présents dans `tab`.
2. Écrire une fonction NOMBREDENOMBRESPAIRS qui prend en paramètre un tableau d'entiers `tab` et qui retourne le nombre de nombres pairs présents dans `tab`. Le code de cette fonction ne doit contenir qu'une seule instruction.
3. Écrire une fonction NOMBREDENOMBRESIMPAIRS qui prend en paramètre un tableau d'entiers `tab` et qui retourne le nombre de nombres impairs présents dans `tab`. Le code de cette fonction ne doit contenir qu'une seule instruction.
4. Écrire une fonction TRIPARITE qui prend en entrée un tableau d'entiers `tab` et qui retourne un tableau contenant les mêmes éléments que `tab` mais ordonnés tels que les nombres pairs soient tous placés avant les nombres impairs.
5. **(challenge)** Écrire une fonction TRIPARITEENPLACE qui prend en entrée un tableau d'entiers `tab` et qui l'ordonne tel que les nombres pairs soient tous placés avant les nombres impairs. Le code de cette fonction ne doit contenir qu'une boucle et ne doit pas créer de nouveaux tableaux.
6. **(challenge ++)** Écrire une fonction TRIMULTIPLEDE3ENPLACE qui prend en entrée un tableau d'entiers `tab` et qui l'ordonne selon le reste modulo 3. Plus précisément, on veut qu'en sortie de la fonction, si $i < j$ alors `tab`$[i] \mod 3 \leq$ `tab`$[j] \mod 3$. Le code de cette fonction ne doit contenir qu'une boucle et ne doit pas créer de nouveaux tableaux.