

Algorithmique des tableaux

Le sujet comporte 6 pages. Aucun document n'est autorisé. Répondez directement sur le sujet.

Exercice 1 Pour chaque valeur de la variable x complétez le tableau suivant :

- en calculant la valeur de l'expression correspondante et
- en précisant les étapes de calcul nécessaires pour obtenir le résultat.

	Expression	Valeur	Justification
$x=8$	$x\%2==0$ or $x>10$		
$x=7$	$x\%2==0$ or $x<10$		
$x=20$	$x>0$ and $x\%3==0$		
$x=15$	$x\%2==0$ and $x\%3==0$		
$x=4$	$\text{not}(x<0)$ or $\text{not}(x\%2==0)$		
$x=10$	$\text{not}(x\%2==0$ or $x\%3==0)$		
$x=4$	$\text{not}(x\%2==0$ and $\text{not}(x\%3==0))$		

Exercice 2 On suppose que l'on dispose d'une fonction `Pile()` qui simule un lancer d'une pièce. Si le résultat du lancer est *pile*, la fonction renvoie `True`. Si c'est *face*, elle renvoie `False`.

1. Écrire une fonction `lancer(k)` qui renvoie le nombre de lancers nécessaires (le nombre d'appels à la fonction `Pile()`) pour que le nombre de *pires* moins le nombre de *faces* soit au moins k . On supposera $k > 0$ entier.

Par exemple, si la suite des lancers de *pires* (P) ou *faces* (F) est

F P F P P F P P ...

alors `lancer(2)` devra renvoyer 8. Notez que sur la même suite de *pires* ou *faces*, `lancer(1)` aurait renvoyé 5.

2. On souhaite modifier la fonction `lancer(k)` de façon à obtenir le nombre de lancers nécessaires pour que le nombre de *pires* moins le nombre de *faces* soit au moins k et que le nombre de *pires* soit un multiple de k . On supposera toujours $k > 0$ entier. Donner les modifications à apporter à la fonction de la question précédente.

Exercice 3 On définit la factorielle impaire d'un entier $n > 0$ comme le produit de tous les entiers positifs impairs inférieurs ou égaux à n . De plus, par convention, la factorielle impaire de 0 vaut 1.

1. Compléter (sans effectuer le calcul) :

La valeur de la factorielle impaire de 7 est : $1 \times 3 \times \dots$

La valeur de la factorielle impaire de 10 est : $1 \times 3 \times \dots$

Nom:

Prénom:

Groupe:

2. Ecrire une fonction `factorielleImpaire(n)` qui calcule et renvoie la valeur de la factorielle impaire de n . On suppose $n \geq 0$.

3. On souhaite écrire une fonction `sommeFactoriellesImpaires(n)` qui calcule et renvoie la somme $factorielleImpaire(0) + factorielleImpaire(1) + \dots + factorielleImpaire(n-1) + factorielleImpaire(n)$. Écrivez une première version de `sommeFactoriellesImpaires(n)` en utilisant la fonction `factorielleImpaire` écrite précédemment.

4. Combien de multiplications sont nécessaires pour calculer `sommeFactoriellesImpaires(n)`? Justifier.

5. Écrivez une deuxième version `sommeFactoriellesImpairesBis(n)` en modifiant la version précédente pour calculer le résultat **sans** utiliser d'appel à la fonction `factorielleImpaire`.

6. Combien de multiplications sont nécessaires pour calculer `sommeFactoriellesImpairesBis(n)`? Justifier.

Exercice 4 Soit un tableau t de n entiers. On dit qu'un élément $t[i]$, $0 < i < n - 1$ est un **pic** si et seulement si $t[i - 1] < t[i] > t[i + 1]$.

Ainsi par exemple, pour le tableau $t = [7, 8, 5, 6, 4, 10, 9]$, l'élément $t[1] = 8$ est un **pic** car $7 < 8$ et $8 > 5$.

1. Ecrire une fonction `estPic(t, n, i)` qui vérifie si l'élément du tableau à l'indice i est un **pic**. Le tableau t , son nombre n d'éléments et un indice i sont passés en paramètre. Pour l'exemple ci-dessus, l'appel à la fonction `estPic(t, 7, 3)` renvoie `True`, et l'appel à la fonction `estPic(t, 7, 4)` renvoie `False`. Si i vaut 0 ou $n - 1$, la fonction doit renvoyer `False`.

Nom:

Prénom:

Groupe:

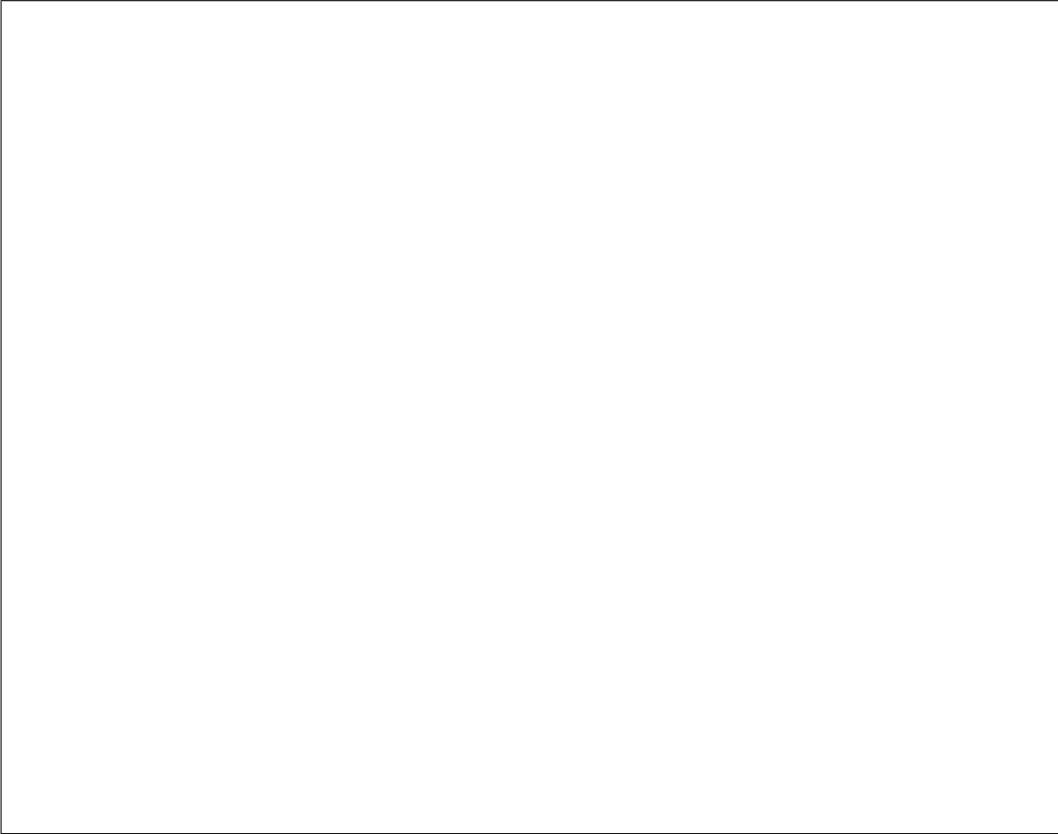
2. Ecrire une fonction `nombreDePics(t, n)` qui renvoie le nombre de pics dans le tableau (contenant n éléments) passé en paramètre.

Pour l'exemple ci-dessus, l'appel à la `nombreDePics(t, 7)` va renvoyer 3

Exercice 5 Étant donné deux tableaux s et t , on dit que s est un sous-tableau de t quand la séquence de valeurs dans s est une sous-séquence de celle dans t . Plus précisément, si $s = [s_0, s_1, \dots, s_m]$ et $t = [t_0, t_1, \dots, t_n]$, s est un sous-tableau de t quand $m \leq n$ et il existe une suite d'indices $i_0 < i_1 < \dots < i_m \leq n$ tels que $s_0 = t_{i_0}, s_1 = t_{i_1}, \dots, s_m = t_{i_m}$.

Par exemple, $[5, 3, 6]$ est un sous-tableau de $[4, 1, \mathbf{5}, 5, 6, \mathbf{3}, 2, 9, \mathbf{6}, 6, 4]$ et $[2, 5, 8, 9]$ est un sous-tableau de $[1, 5, \mathbf{2}, \mathbf{5}, 9, \mathbf{8}, 1, 1, \mathbf{9}, 2]$.

1. Écrire une fonction `sousTableau(s,m,t,n)` qui prend en entrée deux tableaux s et t contenant respectivement m et n éléments. La fonction devra retourner `True` quand s est sous-tableau de t et `False` sinon.



2. On utilise comme mesure de complexité le nombre de tests d'égalité effectués entre une valeur venant de s et une valeur venant t . Évaluez ce nombre (en fonction de m et n) pour votre fonction.

