Série semaine 1

1 Que font ces algorithmes ?

On considère les deux algorithmes suivants :

algorithme 1
entrée : *liste L de nombres entiers, de taille n*
sortie : *variable binaire r (oui/non)*

```
r ⟵ oui
Pour i allant de 2 à n
    Si L(i − 1) > L(i)
        r ⟵ non
Sortir : r
```

algorithme 2
entrée : *liste L de nombres entiers, de taille n*
sortie : *nombre entier positif d*

```
Si n = 1
    Sortir : 0
d ⟵ |L(2) − L(1)|
Pour i allant de 3 à n
    Si |L(i) − L(i − 1)| > d
        d ⟵ |L(i) − L(i − 1)|
Sortir : d
```

a) Quel est la sortie de chacun de ces algorithmes si les données en entrée valent $L = (7, 14, 22, 13, 17)$ et $n = 5$?

b) De manière générale, quelle est la sortie de chacun de ces algorithmes ?

2 Quel est le bon algorithme ?

Lequel des quatre algorithmes suivants permet de calculer la somme des n premiers nombres pairs ? (exemple : si $n = 4$, alors s doit valoir $2 + 4 + 6 + 8 = 20$) Expliquez pourquoi les autres ne fonctionnent pas !

algorithme 1
entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

```
s ⟵ 0
Pour i allant de 1 à n
    Si i est pair
        s ⟵ s + i
Sortir : s
```

algorithme 2
entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

```
s ⟵ 0
Pour i allant de 1 à n
    s ⟵ s + 2i
Sortir : s
```

algorithme 3
entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

```
s ⟵ 0
Pour i allant de 1 à 2n
    s ⟵ s + i
Sortir : s
```

algorithme 4
entrée : *nombre entier positif n*
sortie : *nombre entier positif s*

```
Pour i allant de 1 à n
    s ⟵ s + i
Sortir : 2s
```

3 Réparez-moi ces algorithmes !

Un enseignant du cours ICC a demandé à Jeanne et à Jean d'écrire un algorithme qui calcule la somme des n premiers nombres entiers faisant partie de la liste des multiples de 5 ou de celle des multiples de 7 (ou non-exclusif). Voici ce que Jeanne et Jean ont écrit :

algorithme de Jeanne
entrée : *nombre entier positif* n
sortie : *nombre entier positif* s

$s \longleftarrow 0$
$j \longleftarrow 1$
Pour *i allant de 1 à n*
  Tant que *j n'est ni un multiple de 5 ni un multiple de 7*
    $j \longleftarrow j + 1$
  $s \longleftarrow s + j$
Sortir : s

algorithme de Jean
entrée : *nombre entier positif* n
sortie : *nombre entier positif* s

$s \longleftarrow 0$
$i \longleftarrow 1$
$j \longleftarrow 1$
Tant que $i \leq n$
  $j \longleftarrow j + 1$
  Si *j est un multiple de 5*
    $s \longleftarrow s + j$
    $i \longleftarrow i + 1$
  Si *j est un multiple de 7*
    $s \longleftarrow s + j$
    $i \longleftarrow i + 1$
Sortir : s

Malheureusement, chacun des algorithmes ci-dessus a un problème (différent). Dans chacun des cas, voyez-vous lequel ? Et pouvez-vous aider Jeanne et Jean à réparer leurs algorithmes respectifs ?

Note : Pour identifier concrètement si un nombre j est un multiple de a, on vérifie si $j \bmod a = 0$, où $j \bmod a$ désigne le reste de la division euclidienne de j par a.

4 Ecrivez un algorithme

Soit L une liste de nombres entiers, de taille n, et x un nombre entier positif.

a) Ecrivez un algorithme dont la sortie soit oui s'il existe $i, j \in \{1, \ldots, n\}$ tels que $|L(i) - L(j)| > x$, et non dans le cas contraire.

Note : Il existe plusieurs façons d'écrire un tel algorithme !

b) Estimez maintenant l'efficacité de votre algorithme : combien d'opérations seront effectuées par celui-ci (approximativement) si la taille de la liste L vaut $n = 1'000$?

5 Palindrome

On considère l'algorithme suivant :

algorithme
entrée : *liste L de caractères, de taille n* sortie : *booléen s*

$s \longleftarrow$ *Vrai*
Pour *i allant de* 1 *à* $\lfloor n/2 \rfloor$
    Si $L(i) \neq L(n-i+1)$
        $s \longleftarrow$ *Faux*
Sortir : s

Pour quelle liste de lettres en entrée la sortie de l'algorithme ci-dessus est-elle `Vrai` ?

- [ ] `MALAYALA`
- [ ] `CHACHA`
- [ ] `HAHHAH`
- [ ] `ALABAMA`

6 Somme spéciale

On considère l'algorithme suivant :

somme spéciale
entrée : *nombre entier n* sortie : *nombre entier $total$*
$total \longleftarrow 0$ ***Pour*** *i allant de* 0 *à* $n-1$ *par pas de* 2 $total \longleftarrow total + i$ **Sortir :** $total$

Quelle est la sortie de l'algorithme pour $n = 10$?

- [ ] 8
- [ ] 10
- [ ] 20
- [ ] 30