

Tout genre de support est autorisé pendant l'épreuve.

1. (10 points) On considère l'algorithme suivant :

Algorithme 1: Algo1(x, y, z : Entier)

```

1: Entrée :  $x, y$ 
2: PréC : Rien
3: Sortie :  $z$ 
4: PostC :  $\{?\}$ 
5:  $z \leftarrow 0$ 
6:  $I \leftarrow 0$ 
7:  $s1 \leftarrow 1$ 
8:  $s2 \leftarrow 1$ 
9: Si  $x < 0$  Alors
10:    $s1 \leftarrow -1$ 
11: Si  $y < 0$  Alors
12:    $s2 \leftarrow -1$ 
13: Tant que  $(I < s2 \times y)$  faire
14:    $z \leftarrow z + (s1 \times s2) \times x$ 
15:    $I \leftarrow I + 1$ 

```

1. Quelle est la valeur retournée par l'algorithme (la valeur de z) dans les cas suivants :
 - (a) $x = 0, y = 0$
 - (b) $x = 0, y = 2$
 - (c) $x = 2, y = 0$
 - (d) $x = 2, y = 2$
 - (e) $x = -2, y = 2$
 - (f) $x = -2, y = -2$
 - (g) $x = 2, y = -2$
 2. Combien d'opérations de comparaison effectuées par cette algorithme en fonction des données d'entrée?
 3. Que fait cet algorithme ? En déduire la post-condition (PostC)?
 4. Écrire un algorithme récursif équivalant à cet algorithme?
2. (10 points) L'objectif de cette exercice est de définir un modèle pour une structure chaînée particulière. On veut implémenter une liste de manière à ce que la tête de la liste ainsi que l'élément du milieu soient accessibles directement via des pointeurs. Pour faciliter l'usage de cette liste on doit aussi garder le nombre d'éléments de la liste dans une variable que l'on appelle N (voir Figure 1)
- De plus, cette liste doit être toujours triée dans l'ordre croissant de ces éléments (pour simplifier on suppose que les éléments sont de type Entier). Les opérations doivent donc respecter cette contrainte.
1. Écrire le modèle de données correspondant à cette structure
 2. Écrire un algorithme correspondant à chacune des opérations suivantes (essayer d'exploiter la structure de la liste pour optimiser vos algorithmes) :
 - (a) Create : Entier \rightarrow Liste
Cette opération crée une liste qui contient un seul élément envoyé comme paramètre d'entrée
 - (b) Add : Liste \times Entier \rightarrow Liste
Cette opération ajoute un élément à une liste en gardant la liste triée. Il va donc mettre l'élément à la bonne position. Par exemple, si $L = \langle 2, 4, 6 \rangle$ et l'élément à ajouter $E = 5$, l'algorithme va retourner la liste $\langle 2, 4, 5, 6 \rangle$

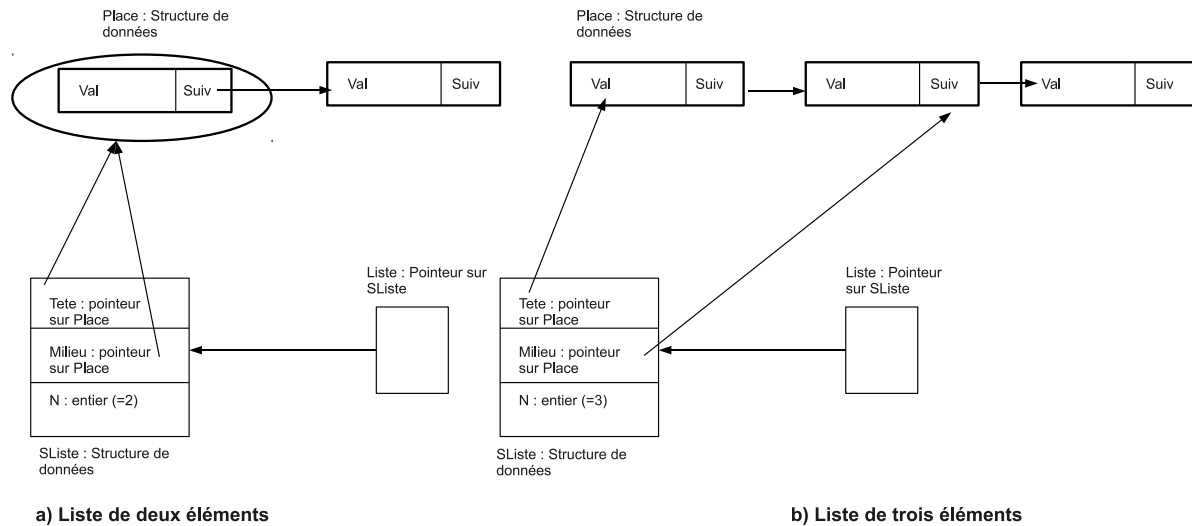


Figure 1: Représentation sémantique du modèle demandé

- (c) Delete : $Liste \times Entier \rightarrow Liste$
 Cette opération supprime la première occurrence d'un élément E dans une liste
- (d) Serach : $Liste \times Entier \rightarrow Booléen$
 Cette opération cherche si un élément E existe dans une liste L . Il retourne **vrai** si l'élément existe, sinon il renvoi **faux**.