

INFO510 – CT SESSION 1

Documents autorisés : une feuille manuscrite au format A4.

Directives :

- Tout le code demandé doit être rédigé dans le langage pseudo-code vu en cours.
- Vous pouvez utiliser sans les redéfinir tous les types de données abstraits vu en cours et en TD.
- Définir des fonctions intermédiaires, afin d'améliorer la lisibilité et compréhension de vos algorithmes sera valorisé.
- Le barème est donné à titre indicatif. Des points négatifs seront attribués à la présentation.

Exercice I. Trace d'un programme (/4)

Déterminez ce qui est affiché par le programme ci-dessous :

Procédure $u(x : \underline{\mathbf{ES}}$ Entier, $y : \underline{\mathbf{E}}$ Chaîne)

Variable : lng : Entier

Début

Affiche ("x =", x , ", y =", y)

$lng := \text{taille}(y)$

Si $lng > 0$ **alors**

$x := x + \text{charToInt}(y)$

$y := \text{sousChaîne}(c, 1, lng - 1)$

$u(x, y)$

Fin si

Fin

Variable : w, z : Entiers initialisés à 0

Début

$u(w, \text{"abc"})$

Affiche ("w =", w)

$u(z, \text{"idees"})$

Affiche ("z =", z)

Fin

Ci-dessous les définitions des deux primitives dont vous aurez besoin :

- Fonction qui renvoie une copie de la chaîne commençant à l'indice pos de longueur lng :
 Fonction $\text{sousChaîne}(c : \underline{\mathbf{E}}$ Chaîne, $pos : \underline{\mathbf{E}}$ Entier, $lng : \underline{\mathbf{E}}$ Entier) : Chaîne
- Fait correspondre à chaque caractère un entier, (avec la convention, *ici*, que a correspond à 1, b correspond à 2, et.)
 Fonction $\text{charToInt}(c : \underline{\mathbf{E}}$ Caractère) : Entier

1) Déterminez ce que le programme affiche. (/2)

2) Est-on sûr que l'action u se terminera pour n'importe quelle entrée? Justifier. (/2)

Exercice II. Tableaux et complexité (/6)

Etant donné un tableau T de n entier, on veut savoir s'il existe un élément majoritaire, c'est-à-dire un élément dont le nombre d'occurrences (c'est à dire le nombre de fois où l'élément apparaît) est strictement supérieur à $n/2$, et dans ce cas, le déterminer.

L'algorithme naïf consiste à vérifier pour chaque élément du tableau s'il est majoritaire.

1. Ecrire l'en-tête de la fonction ou procédure (/1)
2. Compléter le corps de celle-ci pour réaliser l'algorithme naïf (/2)
Dans le cas où il n'exite pas d'élément majoritaire, votre fonction renverra -1
3. Quel est la complexité de votre algorithme? Justifier (/3).
Une analyse correcte, même si la question précédente n'est pas correcte, rapportera des points

Exercice III. Implémentation d'un type abstrait : Bag ou Multi-ensemble (/10)

Le type abstrait *Bag*, (ou *Multi-ensemble*) est un type abstrait utilisé en informatique : on le trouve notamment implémenté dans les bibliothèques standards de Java (HashBag ou TreeBag) de Python (collections.Counter), etc...

Le *Bag* est un type abstrait où l'on peut ajouter autant de fois que l'on veut la présence d'un élément.

Exemple d'exécution

```
Variables : B : Bag
Debut
  initialiserBag(B)
  ajoute(B, 4)
  ajoute(B, 3)
  ajoute(B, 7)
  afficher(B)
  ajoute(B, 4)
  ajoute(B, 4)
```

```
ajoute(B, 3)
supprime(B, 7)
afficher(B)
supprime(B, 4)
afficher(contient(B, 4))
afficher(compte(B, 4))
afficher(contient(B, 7))
afficher(compte(B, 7))
Fin
```

Le programme affichera :

```
4, 3, 7
4, 4, 4, 3, 3
True
2
False
0
```

L'objectif de cet exercice est d'implémenter les différentes primitives de *Bag*, sur un type : **Element**, défini comme un entier.

Type : Element : Entier

Primitives sur les Bag

Ci-dessous les différentes primitives sur les Bag :

- Permet d'initialiser un Bag
Procédure initialiserBag($B : \underline{\mathbf{S}}$ Bag)
- Permet d'ajouter un élément dans un Bag
Ajouter un élément déjà présent est possible : l'élément sera présent une fois de plus
Procédure ajoute ($B : \underline{\mathbf{ES}}$ Bag, $x : \underline{\mathbf{E}}$ Element)
- Permet de savoir si un élément est présent ou non (peu importe le nombre de fois)
Fonction contient($u : \underline{\mathbf{E}}$ Bag, $x : \underline{\mathbf{E}}$ Element) : Booléen
- Permet de compter le nombre de fois qu'un élément est présent
Fonction compte ($B : \underline{\mathbf{E}}$ Bag, $x : \underline{\mathbf{E}}$ Element) : Entier
- Permet de supprimer un élément
Procédure supprime ($B : \underline{\mathbf{E}}$ Bag, $x : \underline{\mathbf{E}}$ Element)

Questions

- 1) Définissez le type abstrait *Bag*, en recopiant et complétant le début de définition ci-dessous.
(/2)

Cette représentation sera celle que vous garderez pour la suite de l'exercice.

Type : Bag : Entité

tabElements : Tableau de MAX Eléments

...

- 2) Implémentez les 5 primitives ci-dessus. (/5)
- 3) Une des utilisations peut être de vérifier qu'un multi-ensemble est contenu dans un autre : chaque élément du premier doit être présent dans le second au moins le même nombre de fois. Implémentez *estSousBag* (n'oubliez pas d'écrire l'en-tête).(/3)