Méthodologie de la Programmation

1ère année Apprentissage
18 Décembre 2020.
Durée 1h30
Documents de cours Autorisés

Tous les programmes demandés **seront écrits en langage algorithmique ou bien an Ada.**
Ces programmes devront respecter scrupuleusement **TOUTES les consignes de bonne programmation définies en cours, TD et TP**

.

1 Questions de cours

C.1. Y a-t-il des liens entre méthode des raffinages et sous-programmes ?

C.2. Y a-t-il des liens entre raffinages et architecture logicielle ?

C.3. Qu'est ce qu'un module générique ? Quel est son intérêt ?

C.4. A quoi sert une précondition ? Une postcondition ?

C.5. Comment différenciez-vous programmation offensive et programmation défensive ?

2 Trouver les erreurs

Considérons les entêtes (signatures) des deux sous-programmes suivants.

```
PROCEDURE F(X : IN OUT ENTIER)

PROCEDURE G(X : IN OUT ENTIER ; N : IN ENTIER)
```

Soient `a`, `M`, `P` et `Y` des variables de type `ENTIER`, déjà initialisées, et considérons la liste suivante des appels à ces sous-programmes

```
1. G (P, 3)

2. Y := G(Y, M);

3. G (P, 3+X);

4. G (Y*Y, M);

5. G (Y, F(M));
```

Question

E.1. Pour chacun des appels précédents, déterminer si cet appel est correct. Les réponses doivent être justifiées.

3 Conception par raffinage

Dans cette partie, nous nous intéressons au problème de l'élimination d'éléments redondants dans un tableau.

Pour la suite, on considérera des tableaux de caractères déclarés sous-la forme suivante.

```
CONSTANTE Capacite : ENTIER := 10

TYPE T_Tabl_Caract EST TABLEAU(1..Capacite) DE Caractere

TYPE T_Tab EST ENREGISTREMENT de
 Le_Tab : T_Tabl_Caract
 Taille : Entier     -- Taille >=0 ET Taille <= Capacite
FIN ENREGISTREMENT
```

On dispose d'un tableau T de caractères à une dimension non-totalement rempli. Le tableau peut contenir plusieurs occurrences d'une même valeur.

On souhaite définir un sous-programme ELIMINE permettant de ne conserver dans le tableau T qu'une seule occurrence d'une même valeur.

Pour résoudre ce problème, on n'utilisera pas de fonction intermédiaire ni de tableau intermédiaire.

A- Elimination des éléments redondants dans un tableau NON TRIÉ

L'objectif est donc d'éliminer les éléments redondants dans un tableau, qui **N'EST PAS TRIÉ**.

Pour cette question, on **NE TRIERA PAS** le tableau au préalable. L'exemple ci-dessous montre l'utilisation du sous-programme demandé dans le cas d'un tableau qui n'est pas trié.

Le tableau suivant

`Taille =8`

1	2	3	4	5	6	7	8	9	10
'b'	'a'	'a'	'b'	'f'	'f'	'e'	'f'		

devient le tableau

`Taille =4`

1	2	3	4	5	6	7	8	9	10
'b'	'a'	'f'	'e'						

A1. Spécifier un sous-programme qui, à partir d'un tableau donné, fournit, ce même tableau, dans lequel une seule occurrence d'un élément initial est présente dans ce tableau. Les éléments ne sont plus redondants.

A2. En utilisant la méthode des raffinages, concevoir un algorithme qui implante la spécification issue de la question A1. On décrira les différentes étapes de raffinement ainsi que les différents sous-programmes qui résulteraient de ces raffinages

A3. Comment faire pour généraliser l'algorithme précédent à des types d'éléments quelconques (par exemple des réels, des complexes, etc.) ?

B - Et si le tableau était préalablement trié !

Ici, on supposera que le tableau est **PREALABLEMENT TRIÉ**

L'exemple ci-dessous montre l'utilisation du sous-programme demandé dans le cas d'un tableau TRIÉ.

Le tableau suivant

`Taille =8`

1	2	3	4	5	6	7	8	9	10
'a'	'a'	'a'	'b'	'c'	'c'	'c'	'e'		

devient le tableau

`Taille =4`

1	2	3	4	5	6	7	8	9	10
'a'	'b'	'c'	'e'						

Par rapport au cas précédent où le tableau en entrée n'est pas trié, décrire succinctement comment sont modifiés

B1. la spécification de ce sous-programme ;

B2. le raffinage de ce sous-programme.