

TD - EXERCICES DE REVISION

Exercice 1 : Généralité

Exposer la structure générale d'un programme en ada (Où déclarer les types ? Où écrire le corps des sous-programmes ? etc... ?)

Exercice 2 : Généralité

En ada, il existe deux « familles » de sous-programmes : les **procédures** et les **fonctions**.

Question 1 : Expliquer quels sont les différents **modes de passages de paramètres** possibles, ceci pour ces 2 « familles » de sous-programmes. Appuyez-vous sur des exemples de spécifications de sous-programmes.

Question 2 : Montrer comment « utiliser » ces différents sous-programmes.

Exercice 3 : Généralité

Vous disposez d'une procédure d'affichage de tableau :

```
procedure Afficher(T : in Tableau);
```

avec le type tableau défini ainsi :

```
type Tableau is array (Integer range <>) of Character ;
```

Ecrire le programme de test de cette fonction.

Exercice 4 : Tableaux contraints et non-contraints

Question 1 : Ecrire un programme ada avec :

- La déclaration d'un type tableau **contraint** d'entiers naturels indicé par des entiers relatifs.
- Une procédure qui calcul la somme des valeurs du tableau (en itératif).
- La déclaration d'une variable de type tableau **sans l'initialiser**.
- Un programme principal qui :
 - o Initialise la variable tableau avec quelques valeurs.
 - o Demande le calcul de la somme des valeurs du tableau et affiche le résultat.

Question 2 : Refaire le même programme en déclarant un type tableau **non-contraint**. Changer également la procédure de calcul de la somme en une fonction.

Question 3 : Dans quel cas (tableau **contraint** ou **non-contraint**) est-il envisageable de demander la taille du tableau à l'utilisateur ? Pour le(s) cas possible(s), et en considérant que l'on dispose d'une procédure `saisie(T : out Tableau)` de saisie des valeurs du tableau par l'utilisateur, modifier le programme principal afin que l'utilisateur saisisse le tableau.

Exercice 5 : Exceptions

Question 1 : Où et comment déclarer une exception ?

Question 2 : Où et comment déclencher une exception ?

Question 3 : Où et comment traiter une exception ?

Question 4 : La procédure `Afficher_Somme(T)` du programme `Exercice5` en annexe lance la fonction `Calculer_Somme(T)` (pour calculer la somme des valeurs d'un tableau) puis affiche le résultat. On constate que lorsque un tableau vide est donné en paramètre, il est affiché que la somme vaut 0. En fait, cette somme n'existant pas, puisqu'il n'y a pas de valeur.

Améliorer ce programme en utilisant une exception `Exception_Tableau_Vide`.

Exercice 6 : Récursivité

Question 1 : Ecrire une fonction récursive qui recherche la valeur la plus grande dans un tableau d'entiers naturels indicé par des entiers relatifs.

Question 2 : Quel(s) changement(s) faudrait-il apporté pour un tableau d'entiers relatifs (toujours indicé par des entiers relatifs) ?

Question 3 : Modifier le sous-programme pour en faire une procédure.

Exercice 7 : Pointeurs

Soient les définitions des types suivantes :

```
Type Cellule ;  
Type Liste is access Cellule ;  
Type Cellule is record  
    Info : Integer ;  
    Suiv : Liste;  
end record;
```

Question 1 : Ecrire une procédure d'affichage de liste en itératif, puis en récuratif.

Question 2 : Ecrire une procédure d'insertion de valeur dans une liste ordonnée en itératif, puis en récuratif.