

TD6 TEC 611 1991 algorithmes paramétrés , définitions et appels

EX1- soit l'algorithme suivant :

```
procedure <premier>  
entier) ;
```

```
debut
```

```
I, J : entiers ;
```

```
I ← 1 ; J ← 2*I ;
```

```
permuter (I,J); afficher(I ,J)
```

```
fin <premier>
```

```
procedure <permuter> (I,AUX :
```

```
debut
```

```
J : entier;
```

```
J ← 2;
```

```
AUX ← I ; I ← J ; J ← AUX;
```

```
fin <permuter>;
```

a. que fait l'algorithme <premier> ?

b. si on remplace l'entête de la procédure <permuter> par :

procedure <permuter> (var I , AUX : entier) ; quel serait l'effet de <permuter > dans l'algorithme premier ?

c. on veut pouvoir permuter n'importe quelle paire de valeurs lues à partir du clavier. les modifications suivantes permettent –elles de réaliser cela ? sinon porter les modifications nécessaires afin de réaliser cette tâche.

```
procedure <premier>
```

```
var I,AUX : entier) ;
```

```
debut
```

```
I , J : entier;
```

```
lire(I) ; lire(J) ;
```

```
AUX;
```

```
permuter (I,J); afficher(I ,J)
```

```
fin <premier>;
```

```
procedure <permuter> (
```

```
debut
```

```
J ← 2;
```

```
AUX ← I ; I ← J ; J ←
```

```
fin <permuter>;
```

EX2-on considère un processeur pour lequel on ne peut faire plus d'une addition par action d'affectation numérique.

soit l'algorithme paramétré suivant :

```
procedure <SOMME1> (A, B, C, D, S) ;
```

```
debut
```

```
S ← 0;
```

```
S ← S+A;
```

```
S ← S+B;
```

```
S ← S+D;
```

```
fin <SOMME1>;
```

a. que réalise cette procédure ?

b. utiliser cette procédure pour faire la somme de trios variables X, Y, Z dans la variable TOTAL.

c. si K, L, M et N sont 4 variables numériques, quel sera l'effet de l'appel SOMME1(K, L, M, N, K). donner une séquence d'actions utilisant un appel de SOMME1 et permettant de faire la somme des valeurs de K, L, M, et N dans K.

d. composer pour le même processeur un algorithme SOMME2(X, Y, Z, T) qui calcule dans le paramètre-donnée-résultat X la somme des valeurs de X, Y, Z, et T , (Y, Z, et T étant les paramètres données).

- e. donner une séquence d'actions utilisant l'algorithme SOMME2 et calculant la somme des six(6) variables N1, N2, N3, N4, N5, et N6 dans la variable CUMUL.
- f. peut-on utiliser SOMME1 ou SOMME2 pour quadrupler la valeur d'une variable ?

EX3-

Quand un ordinateur lit les lignes successives d'un programme écrit en Pascal, sa première tâche est de reconnaître les mots réservés, les identificateurs et les opérateurs du langage. Écrire une procédure, appelée **<examen>**, qui exécute cette tâche. Naturellement, cette procédure peut elle-même appeler d'autres qui lui sont locales et qui exécutent des tâches bien particulières.

A chaque appel, **<examen>** doit lire le mot suivant (ou opérateur suivant) du texte et retourner ce mot comme valeur d'une variable globale appelée **MotExamine**. Le type de **MotExamine** sera `packed array[1..80] of char`.

Une autre variable globale appelée **LongeurMot**, de type integer, contiendra le nombre de caractères du mot examiné.

Les mots réservés et les opérateurs du langage auront des numéros, et **<examen>** devra affecter le numéro approprié à la variable globale **Indice**. Cette variable aura la valeur :

- 1 pour un identificateur,
- 2 pour une constante numérique,
- 3 pour une constante booléenne,
- et 4 pour une constante caractère.

Aux symboles illégaux (comme 12.ABC) on associera la valeur 0.

Le problème sera considérablement simplifié par la remarque suivante :
Aucun mot ne pourra être coupé en fin de ligne. Ne pas oublier, enfin d'abandonner la procédure **<examen>** pour un début et fin de commentaire.