

PROGRAMMATION 1

TD4

Notions à acquérir

Les procédures

Une procédure est une **abstraction d'instruction**. C'est en fait une suite d'instructions qui sont regroupées sous un même titre appelé **identificateur de la procédure** et qui échange des données avec la procédure appelante (éventuellement programme) grâce aux **paramètres**.

Une procédure se présente selon le schéma suivant :

procédure *Identificateur (liste des paramètres)*

- **pré-requis** *conditions qui doivent impérativement être vraies lors de l'utilisation de cette procédure*
- **action** *indiquer clairement ce que fait cette procédure, ainsi que le rôle de chaque paramètre*
- **stratégie** *techniques de calcul particulières à cette procédure*
- **variables** *liste des variables locales de la procédure. Leur type et leur rôle dans la procédure y sont explicités*

début

liste des instructions

fin

La liste des paramètres peut être vide.

Distinction entre **variable globale** et **variable locale**

(avec **interdiction** d'utiliser les **variables globales** dans les procédures).

Exercice 1

Ecrire une procédure MULTIPLICATION ({D} N : entier) qui permet d'afficher à l'écran la table de multiplication de 1 à 10 du nombre N, ayant la forme donnée par l'exemple suivant.

par exemple si n=4 :

```

1 * 4 = 4
2 * 4 = 8
3 * 4 = 12
4 * 4 = 16
5 * 4 = 20
6 * 4 = 24
7 * 4 = 28
8 * 4 = 32
9 * 4 = 36
10 * 4 = 40

```

NB : Dans cet exercice, le plus important ce ne sont pas les instructions, mais l'en-tête et les spécifications.

Exercice 2

Ecrire une procédure MIN_MAX ({R} min : entier ; {R} max : entier) qui saisit des nombres entiers positifs à partir du clavier et calcule simultanément la plus petite et la plus grande valeur contenue dans cette suite.

La lecture de la suite de nombres se termine lorsque le nombre -1 est tapé.

Exercice 3

On considère la procédure Ordonne prenant 2 paramètres entiers X et Y en entrée, et qui écrit dans X le plus petit de ces 2 entiers, et dans Y le plus grand de ces 2 entiers.

- Quel est le mode de transmission des paramètres X et Y ?
- Ecrire le code de la procédure Ordonne
- A l'aide (uniquement) de la procédure Ordonne, écrire la procédure suivante :
Procédure Tri3 ({D/R} A, B, C : entier) qui écrit dans A, B, C les valeurs des paramètres triées (on souhaite donc qu'après l'appel de Tri3 on ait $A \leq B \leq C$).

Exercice 4

- Corriger le programme ci-dessous en ajoutant des paramètres aux procédures pour qu'il n'y ait plus d'utilisation des variables globales dans les procédures.
- Faire la trace (commentée) du programme obtenu après les corrections de la question a).
- Quelle est l'action réalisée par la procédure Aux2 ?

programme Casse-Tête

procédure Aux ({D} T : entier)

début

Afficher("hello aux!");

Passer à la ligne

$X \leftarrow X + T$

Afficher(X);

Passer à la ligne

fin Aux

procédure Aux2

variable

X : entier

début

$X \leftarrow 0$

fin Aux2

procédure Aux3

début

$X \leftarrow 0$

fin Aux3

variable

X : entier

début

X ← 0

Aux(1)

Aux(2)

Aux2

Aux(X)

Aux3

Aux(X+3)

fin Casse-Tête

Exercice 5

a) Faire la trace du programme suivant, et écrire les actions du programme et de la procédure :

programme MonProgA

action : ...

procédure Proc1

action : ...

variable

I : entier

début

pour I dans 1..K

faire Afficher (I)

fin pour

fin Proc1

variable :

I, K : entier

début

I ← 15; K ← 4

Afficher (I)

Proc1

Afficher (I)

fin MonProgA

b) La procédure Proc1 ci-dessus utilise la variable globale K de la procédure MonProgA; vous avez vu en cours que c'est interdit : Rappelez pourquoi.

c) Réécrire ce programme sans utilisation de la variable globale dans Proc1 (et refaire la spécification de Proc1).

Exercice 6

Soit la procédure suivante :

procédure TOUT_SIMPLE ({D} NB : entier ; {R} VALEUR : entier)

pré-requis : aucun

action : question c) de cet exercice

variable :

TROUVE : booléen

début

```
VALEUR ← NB
TROUVE ← faux

tant que VALEUR <= 25 et non TROUVE
faire si VALEUR == 25
    alors TROUVE ← vrai
    sinon VALEUR ← VALEUR + 1
fin si
fin Tant que
```

fin

- Donner la condition de sortie de l'itération
- Faire la trace de la procédure avec $NB = 23$, puis avec $NB = 32$
- Ecrire le libellé de la spécification "action" (constituant la deuxième rubrique des spécifications externes de la procédure TOUT_SIMPLE)
- Peut-on réécrire cet algorithme de façon plus simple ? Si oui, donner la solution.

Exercice 7

On souhaite afficher à l'écran une pyramide de hauteur donnée h constituée de suites d'un caractère donné c .

Par exemple, si $h = 5$ et $c = '*'$, on obtient la pyramide suivante :

```
  *
 ***
*****
*****
*****
```

- Ecrire une procédure AFFICHE_SUITE ({D} Nb : entier ; {D} car : caractère) qui permet d'afficher Nb caractères car à partir de la position courante sur l'écran.
- En utilisant la procédure précédente pour afficher des suites d'espaces ou du caractère c , écrire une procédure PYRAMIDE_SIMPLE ({D} h : entier ; {D} c : caractère) qui permet d'afficher à l'écran une pyramide de hauteur h constituée de suites du caractère c .

Exercice 8

Le but de cet exercice est d'afficher une pyramide un peu plus élaborée que la précédente.

- Ecrire une procédure AFFICHE_ESPACE ({D} Nb : entier) qui permet d'afficher Nb caractères ' ' à partir de la position courante sur l'écran.

- b) Ecrire une procédure `AFFICHE_NOMBRES_CROISSANTS` ({D} Nb1, Nb2 : entier) qui permet d'afficher à l'écran sur une même ligne les chiffres représentant les unités des nombres allant de Nb1 à Nb2 en ordre croissant séparés par un espace si $Nb1 \leq Nb2$, et ne fait rien sinon.
Par exemple, si Nb1 = 73 et Nb2 = 75, la procédure affiche 3, 4, 5.
- c) Ecrire une procédure `AFFICHE_NOMBRES_DECROISSANTS` ({D} Nb1, Nb2 : entier) qui permet d'afficher à l'écran sur une même ligne les chiffres représentant les unités des nombres allant de Nb2 à Nb1 en ordre décroissant séparés par un espace si $Nb1 \leq Nb2$, et ne fait rien sinon.
Par exemple, si Nb1 = 73 et Nb2 = 75, la procédure affiche 5, 4, 3.
- d) En utilisant les procédures précédentes, écrire une procédure `PYRAMIDE` ({D} N: entier) qui permet de représenter à l'écran la pyramide suivante si $N = 10$:

```

      1
    2 3 2
  3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
6 7 8 9 0 1 0 9 8 7 6
7 8 9 0 1 2 3 2 1 0 9 8 7
8 9 0 1 2 3 4 5 4 3 2 1 0 9 8
9 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 9
0 1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 0

```

Exercice 9

Les nombres en chiffres romains sont formés selon les règles suivantes :

Chiffre romain	Equivalent en base 10
M	1000
D	500
C	100
L	50
X	10
V	5
I	1

Il n'y a pas de chiffre romain représentant le nombre 0.

La traduction d'un entier compris entre 0 et 3999 de sa représentation usuelle en base 10 à sa représentation en chiffres romains se fait chiffre par chiffre et dans le même ordre (unités à droite).

Le chiffre des unités (de 0 à 9) se traduit respectivement par (rien), I, II, III, IV, V, VI, VII, VIII, IX (noter que le préfixe I devant V et X a un rôle soustractif et non additif).

De même, le chiffre des dizaines (de 0 à 9) se traduit respectivement par (rien), X, XX, XXX, XL, L, LX, LXX, LXXX, XC.

De même, le chiffre des centaines (de 0 à 9) se traduit respectivement par ...

Le chiffre des milliers (de 0 à 3) se traduit respectivement par ...

<i>Exemples :</i>	<i>1999</i>	<i>se représente par</i>	<i>MCMXCIX</i>
	<i>334</i>		<i>CCCXXXIV</i>
	<i>487</i>		<i>CDLXXXVII</i>

1°) Ecrire la procédure suivante :

Procédure CHIFFRE_UNITES ({D} Nb : entier)

-- pré-requis : $0 \leq \text{Nb} \leq 9$

-- action : affiche à l'écran la représentation en chiffres romains, de Nb.

2°) Ecrire la procédure suivante :

Procédure CHIFFRE_DIZAINES ({D} Nb : entier)

-- pré-requis : $0 \leq \text{Nb} \leq 9$

-- action : affiche à l'écran la représentation en chiffres romains de Nb*10.

3°) Ecrire les procédures CHIFFRE_CENTAINES et CHIFFRE_MILLIERS définies de façon analogue aux procédures précédentes.

4°) Ecrire la procédure suivante :

Procédure AFFICHE_EN_CHIFFRES_ROMAINS ({D} Nb : entier)

-- pré-requis : $0 \leq \text{Nb} \leq 3999$

-- action : affiche à l'écran la représentation en chiffres romains de Nb.

5°) Réécrire les procédures précédentes en remplaçant les 4 premières procédures par une unique procédure:

Procédure CHIFFRE ({D} Nb : entier; cUn, cCinq, cDix: caractère)

-- pré-requis : $0 \leq \text{Nb} \leq 9$

-- action : affiche à l'écran la suite de caractères obtenue à partir de la représentation en chiffres romains de Nb en remplaçant I par cUn, V par cCinq et X par cDix.

6°) Ecrire le programme principal CHIFFRES_ROMAINS qui saisit au clavier un entier compris entre 0 et 3999 et affiche à l'écran sa représentation en chiffres romains.