

Rédigé par : Hervé de Milleville

Ref : *ING-MAN1-EXA-ALG*

A l'intention de : Etudiants d'Ingénieur Manager 1A

Créé le : 28/04/2014

Préambule

Cet examen dure 2h00. L'examen se fait sur feuilles. Les documents manuscrits sont autorisés. L'ordinateur en local est autorisé. Vous devez rendre une copie papier. L'examen est assez long, vous serez donc notés sur un total supérieur à 20.

1. Questions de réflexion

- 1) (1pt) Expliquer pourquoi quand on veut intervertir les contenus de 2 variables, on est obligé d'utiliser une variable intermédiaire.
- 2) (2pts) Dans une fonction comme dans une procédure, on peut avoir des paramètres et/ou des variables locales.
 - a. Quelle est la différence essentielle entre un paramètre et une variable locale ?
 - b. Peut-on envisager une fonction ou une procédure n'ayant pas de paramètre ? On peut répondre avec un exemple.
- 3) (2 pts) Procédures versus fonctions
 - a. Rappeler brièvement la différence entre une procédure et une fonction.
 - b. Expliquer comment on peut transformer une fonction en une procédure en utilisant un paramètre particulier.
- 4) (1pt) La forme classique d'une procédure récursive est un SI ALORS cas particulier SINON cas récursif. Il arrive souvent qu'une procédure récursive puisse s'écrire sous forme non récursive. Dans ce cas, on vous demande d'expliquer pourquoi le SI ALORS SINON devient en général une boucle POUR ou un TANT QUE.
- 5) (1pt) Quelle est la différence entre un algorithme et un programme en C ?

2. Traduction algorithmique

2.1 *Minimum et Maximum*

On cherche le maximum et le minimum d'une suite de n nombres ($n \leq 50$) entiers stockés dans un tableau noté t . Pour cela, on impose que ce soit une procédure.

- 1) (1pt) Expliquer pourquoi le prototype (l'entête) de cette procédure peut être :
procedure minMax (t tableau de Entier(50) (E), n : Entier (E), min : Entier (S), max : Entier (S))
- 2) (3pts) Ecrire l'algorithme de cette procédure.

2.2 *Base 10 versus Base 2*

Pour transformer un nombre écrit en base 10 en un nombre écrit en base 2, il faut le diviser autant de fois que possible par 2 (jusqu'à ce que le quotient soit nul) et récupérer les restes de ces divisions en les écrivant du dernier au premier.

Exemple : 15 s'écrit en base 2 comme suit :

$$\begin{array}{ll} 14 \div 2 = 7 & \text{avec un reste qui vaut } 0 \\ 7 \div 2 = 3 & \text{avec un reste qui vaut } 1 \\ 3 \div 2 = 1 & \text{avec un reste qui vaut } 1 \end{array}$$

ING MAN : EXAMEN D'ALGORITHMIQUE

$1 \div 2 = 0$ avec un reste qui vaut 1

Le nombre 15 s'écrit donc en base 2 : 1110 (on a repris les restes ci-dessus du dernier au premier).

Rappel : Si a et b sont deux variables entières (ou constantes entières) alors

- a DIV b donne le quotient de la division de a par b
- a MOD b donne le reste de la division de a par b

(3pts) On vous demande d'écrire la procédure qui reçoit un nombre entier et qui affiche à l'écran ce même nombre en base 2.

Aide : pour être sûr d'écrire la suite de 0 et de 1 dans le bon ordre, on se servira d'un tableau intermédiaire.

2.3 Tri d'un tableau composé de 0, 1 ou 2

On considère un tableau nommé t de n nombres entiers ($n \leq 50$). On suppose que chacun de ces nombres appartient à l'ensemble $\{0, 1, 2\}$. On désire trier ce tableau par ordre croissant.

Pour la procédure, on gèrera deux indices notés ind0 et ind2. Ind0 est initialisé à 0 et ind2 est initialisé à n+1. On parcourt le tableau de la 1^{ère} case à la n^{ième} case et pour chaque case, on incrémente ind0 de 1 si le contenu de la case vaut 0 et on décrémente ind2 de 1 si le contenu de la case vaut 2.

- 1) (1pt) Montrer qu'à la fin du parcours du tableau :
 - ind0 représente l'indice de la dernière case qui vaut 0 dans le tableau trié.
 - ind2 représente l'indice de la première case qui vaut 2 dans le tableau trié.
- 2) (1pt) Expliquer pourquoi l'entête de cette procédure peut être
procédure trierTableau012(t tableau de Entier(50) (ES), n : Entier (E))
- 3) (3pts) Ecrire l'algorithme de cette procédure en gérant des indices ind0 et ind2 comme indiqué ci-dessus.
- 4) (1pt) Montrer que la complexité de cet algorithme est un $O(n)$.
- 5) (1pt) La complexité du meilleur algorithme de tri vu en cours était en $O(n \cdot \log_2(n))$. Pour quelles raisons dans le cas précis de cet algorithme a-t-on réussi à faire mieux ?

3. La méthode du Pivot de Gauss

Dans la méthode vue en cours, nous n'avons pas traité les cas particuliers. Un problème se pose dans la procédure *executerPivotGaussPhase1Etapej*. Nous rappelons ci-dessous, l'algorithme de cette procédure (nous avons mis en gras le problème possible) :

procédure executerPivotGaussPhase1Etapej(A Tableau de Reel(10,10) (ES), B Tableau de Reel(10,10) (ES), n : Entier, i : Entier)

Début

j : Entier

*dilater(A, B, n, **1 / A(i,i)**)*

Pour j ← (i+1) à n pas 1

Transvector(A, B, n, j, i, -A(j,i))

Fin Pour

Fin procédure

Quand " $A(i,i) = 0$ ", la division $1 / A(i,i)$ n'est pas définie. Quand cela est le cas, on cherche dans la colonne i, la première valeur de k supérieure à i telle que $A(k,i) \neq 0$ et on intervertit les lignes n° k et n° i.

- 1) (1pt) En appliquant le principe de décomposition d'un problème en fonctions et procédures, quelle(s) sont (est) les (la) fonction(s) et/ou procédure(s) qu'il faudrait ajouter pour tenir compte de ce problème.
- 2) (3pts) Modifier la procédure *executerPivotGaussPhase1Etapej* en utilisant le(s) fonction(s) et/ou les procédure(s) ci-dessus.

ING MAN : EXAMEN D'ALGORITHMIQUE