IF121: Informatique Fondamentale

Deug MIAS et MASS

Examen du 19 Janvier 2001
Durée: 2 heures
Les documents ne sont pas autorisés.

Toutes les réponses, sauf dans le cas de l'exercice 1, doivent être correctement justifiées. La rigueur du raisonnement, la clarté des explications, mais aussi la qualité de la présentation influeront sensiblement sur la note.
On vous démande d'ajouter des brefs commentaires dans vos programmes, pour expliquer leur fonctionnement.
Le barème sur 20 est donné à titre indicatif et pourra être modifié.

Exercice 1 (Binaires, Invariants, Fonctions, Tables de Vérité (4 points))
Ce premier exercice est composé de 5 questions auxquelles vous devez répondre sans donner de justifications *directement sur la feuille du sujet, qu'il faudra rendre avec votre copie (n'oubliez pas de noter votre nom sur la feuille en haut à droite).*

1: addition binaire *Quel est le résultat de l'opération en base 2 suivante ?*

```
11010 +
11110 +
 1011 =
```

Entourez la bonne réponse:

(1) 100011	*(2) 111111*	*(3) 111011*
(4) 110011	*(5) 1000011*	*(6) 101111*

2: complément à deux *Quelle est la valeur en base 10 du nombre* **11000010** *écrit en complément à 2 sur 8 bits ? Entourez la bonne réponse:*

(1) -62	*(2) -66*	*(3) -67*	*(4) -194*
(5) 62	*(6) 66*	*(7) 67*	*(8) 194*

3: tables de vérité *Complétez la table de vérité suivante:*

A	*B*	*A ou (non B)*
vrai	*vrai*	
vrai	*faux*	
faux	*faux*	
faux	*vrai*	

4: passage de paramètres *Dans un programme supposé correct on rencontre la séquence suivante:*

```
....
LireDonnee(d);
EcrireDonnee(d);
```

On suppose écrites les procédures `LireDonnee` *et* `EcrireDonnee` *qui font ce que leur nom indique. Quelle doit être l'en-tête de chacune de ces procédures ?*

Entourez la meilleure réponse

1. `void LireDonnee(int x)` *et* `void EcrireDonnee(int& y)`

2. `void LireDonnee(int x)` *et* `void EcrireDonnee(int y)`

3. `void LireDonnee(int& x)` *et* `void EcrireDonnee(int y)`

4. `void LireDonnee(int& x)` *et* `void EcrireDonnee(int& y)`

5: Invariants *Etant donné le programme*

```
programme p
variables
i : un entier
debut p
     i <- 0
tant que i < 5 faire
     i ← i + 3
fin tant que
fin p
```

Lesquelles, parmi les propriétés suivantes, sont des invariants de la boucle tant que *?*

a) i est impair *b) i est pair* *c)* $i < 5$

Entourez la bonne réponse:

(1) Aucune	*(2) Seulement a)*	*(3) Seulement a) et b)*	*(4) Seulement c)*
(5) Toutes	*(6) Seulement b)*	*(7) Seulement a) et c)*	*(8) Seulement b) et c)*

Exercice 2 (Matrices symétriques et anti-symétriques (4 points)) *Une matrice carrée m est dite* symétrique *si pour tous i et j, on a*

$$m[i][j] = m[j][i]$$

Elle est dite anti-symétrique *si pour tous i et j on a*

$$m[i][j] = -m[j][i]$$

Par exemple, la matrice suivante est symétrique

1	2	3
2	5	4
3	4	0

et celle-ci est anti-symétrique

0	-2	3
2	0	-4
-3	4	0

Écrivez en pseudo-langage un programme qui:

1. *lit en entrée une matrice carrée de taille n inférieure ou égale à une constante MAX fixée. L'utilisateur devra fournir la taille n de la matrice, et la matrice ligne par ligne.*
2. *vérifie si la matrice est symétrique ou/et anti-symétrique.*

Exercice 3 (Interclassement (6 points)) *Etant données deux séquences triés s_1 et s_2, on peut facilement interclasser leurs éléments pour obtenir une nouvelle séquence triée s_3 qui contient les éléments de s_1 et s_2, comme dans l'exemple suivant, où les séquences contiennent des entiers*

La séquence	*vaut*
s_1	*1 3 5 7 7 9*
s_2	*2 3 12 27*
s_3	*1 2 3 3 5 7 7 9 12 27*

On souhaite pouvoir fusionner deux séquences triées d'entiers qui sont fournies en entrée par l'utilisateur, et dont la longueur ne dépasse pas une constante donnée MAX. Pour représenter les séquences, on utilisera une structure ayant deux composantes:

- *un tableau de type tab, pour contenir les entiers,*
- *un entier qui donne le nombres d'éléments significatifs dans le tableau (en commençant par l'indice 1).*

On supposera donc dans cet exercice déjà déclarés les constantes et types suivants:

```
constante MAX=100
type tab  = tableau de MAX entiers
type seq  = structure
               t: tab
               n: entier
            fin structure
```

1. *Ecrivez, en pseudo-langage, une* fonction est_trié *qui admet en paramètres une séquence s de type seq. Cette fonction retournera le booléen vrai si les éléments de la séquence sont triés en ordre croissant, et faux sinon.*

2. *Ecrivez, en pseudo-langage, une* procédure interclasse *qui interclasse les données de deux séquences s_1 et s_2 dans une séquence s_3. Les séquences s_1 et s_2 sont de type seq, et on vous demande de définir le type de la séquence s_3.*

Faites particulièrement attention à définir correctement les entêtes de vos fonctions et procédures.

Exercice 4 (Invariants (4 points)) *Voici un programme écrit en pseudo-langage.*

```
constante MAX=100
type tab = tableau de MAX entiers
programme tabmult
variables
     t       : tab
     i, m : entiers
debut tabmult
     lire t
     m <- t[1]
     i <- 2
     tant que (i <= MAX) faire
          si t[i] = 0 alors
             m <- 0
             i <- MAX+1
          sinon
             m <- m * t[i]
             i <- i+1
          fin si
     fin tant que
     ecrire m
fin tabmult
```

Montrez à l'aide d'un invariant que le programme affiche le produit des éléments du tableau t.

Exercice 5 (Programmation `C--` (2 points)) *Traduisez en `C--` le programme écrit en pseudo-langage de l'exercice 4.*