e3a

CONCOURS ENSAM - ESTP - ARCHIMEDE

Épreuve d'Informatique MP

durée 3 heures

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, d'une part il le signale au chef de salle, d'autre part il le signale sur sa copie et poursuit sa composition en indiquant les raisons des initiatives qu'il est amené à prendre.

L'usage de la calculatrice est interdit

Indiquer en tête de copie ou de chaque exercice le langage utilisé

Exercice 1

a) Ecrire la fonction

```
Sym  données   M : tableau à deux dimensions
               n : entier
     résultat  x : booléen
```

qui prend en entrée le tableau des coefficients d'une matrice carrée de taille n et renvoie la valeur 1 si la matrice est symétrique et 0 sinon.

b) Ecrire la fonction

`Egalite-listes`	`données`	L : `liste`
		L' : `liste`
	`résultat`	b : `booléen`

qui prend en entrée deux listes et renvoie la valeur 1 si les deux listes sont égales et 0 sinon. Ces listes contiennent des entiers positifs ou nuls et se terminent par la valeur -1.

c) Un tableau d'entiers T est dit *trié par ordre croissant* si ses entrées sont triées par ordre croissant : si l est la longueur du tableau, $\forall i, j \in \{1, ..., l\}, i \leqslant j \Rightarrow T[i] \leqslant T[j]$. Il est dit *trié par ordre décroissant* si ses entrées sont triées par ordre décroissant : si l est la longueur du tableau, $\forall i, j \in \{1, ..., l\}, i \leqslant j \Rightarrow T[i] \geqslant T[j]$. Dans les deux cas, il est dit *trié*. Remarquons qu'un tableau T peut être à la fois trié dans l'ordre croissant et dans l'ordre décroissant, s'il vérifie : si l est la longueur du tableau, $\forall i, j \in \{1, ..., l\}, T[i] = T[j]$. Il alors est dit *constant*.

Ecrire la fonction

`Test-tri`	`données`	T : `tableau d'entiers`
		l : `entier`
	`résultat`	x : `entier`

qui prend en entrée un tableau T de longueur l et retourne la valeur 1 si le tableau T est trié par ordre croissant et non constant, -1 si T est trié par ordre décroissant et non constant, 0 si T est constant et 2 si T n'est pas trié.

Ecrire la procédure

`Fusion`	`données`	T : `tableau d'entiers`
		l : `entier`
		U : `tableau d'entiers`
		m : `entier`
	`résultat`	W : `tableau d'entiers trié`

qui fusionne deux tableaux triés tous deux par ordre croissant (respectivement tous deux triés par ordre décroissant), de longueurs respectives l, m ; le résultat est un tableau trié par ordre croissant (respectivement par ordre décroissant) de longueur $l + m$. Dans le cas où les deux tableaux ne sont pas tous deux triés dans un même ordre, la procédure `Fusion` renverra un tableau vide et un message d'avertissement.

Exercice 2

$a)$ Que retourne le programme suivant :

```
S ← 2007
i ← 1
tant que (i² < S)  faire  i ← i + 1
fin tant que
afficher(i)
```

$b)$ Que calculent les procédures suivantes :

```
TR(N : entier, T : tableau d'entiers de longueur N, l : entier, m : entier)
si (1 ≤ l) et (l ≤ m) et (m ≤ N) faire
                pour (i = 1) à (i = m − l + 1) faire TR[i] ← T[l − 1 + i]
                retourner TR
fin si
```

```
Valeur(N : entier, T : tableau d'entiers de longueur N, l : entier)

si (1 ≤ l) et (l ≤ N) faire
    x ← T[1]
    g ← 0
    d ← N + 1
    j ← 1
    tant que (j ≤ N)  faire
                        si (T[j] < x)  faire  g ← g + 1
                                              U[g] ← T[j]
                        si (T[j] > x)  faire  d ← d − 1
                                              U[d] ← T[j]
                        fin si
              j ← j + 1
    fin tant que
    si (g < l < d)  afficher x
    sinon           si (l ≤ g)  faire Valeur(g, TR(N, U, 1, g), l)
                    si (l ≥ d)  faire Valeur(N − d + 1, TR(N, U, d, N), l − d + 1)
    fin si
sinon retourner "l NON VALIDE"
fin si
```

Exercice 3

Soit n un entier naturel. On dit que n est un nombre parfait si $2n$ est la somme des entiers naturels diviseurs de n. Par exemple, l'entier 6 est un nombre parfait puisque $2 \times 6 = 12 = 6 + 3 + 2 + 1$. Ecrire un programme qui détermine la liste des nombres parfaits n tels que $n \leq 9999$.

Exercice 4

Un entier naturel n étant donné, on calcule le produit $\texttt{prod}(n)$ de ses chiffres dans son écriture en base 10, puis le produit des chiffres de $\texttt{prod}(n)$ dans son écriture en base 10, et on recommence ainsi l'application de `prod` jusqu'à obtenir un chiffre entre 0 et 9. Le nombre minimal de fois où on applique `prod` pour transformer n en un chiffre entre 0 et 9 est appelé la *persistance* de n. Par exemple, la persistance de 9 est égale à 0, celle de 97 est égale à 3, car $\texttt{prod}(97) = 9 \times 7 = 63, \texttt{prod}(63) = 6 \times 3 = 18, \texttt{prod}(18) = 1 \times 8 = 8$, et celle de 9575 est égale à 5, car $\texttt{prod}(9575) = 1575, \texttt{prod}(1575) = 175, \texttt{prod}(175) = 35, \texttt{prod}(35) = 15, \texttt{prod}(15) = 5$. Ecrire un programme qui calcule le plus petit entier naturel de persistance 5.

Exercice 5

On considère l'alphabet à deux lettres: $\Sigma = \{a, \bar{a}\}$.
Soit n un entier naturel non nul. Un buffer de taille n est modélisé par un automate $\mathcal{A}_n = (Q_n, \Delta_n, 0, F_n)$ sur l'alphabet Σ défini par :

- Q_n, l'ensemble des états de $\mathcal{A}_n$ est l'ensemble $\{0, 1, 2, ..., n\}$. L'état i représente la présence de exactement i bits dans le buffer.
- La lettre a représente l'arrivée d'un bit dans le buffer. La lettre $\bar{a}$ représente la sortie d'un bit du buffer.
- Δ_n, est l'ensemble des transitions de $\mathcal{A}_n$:
  - $\forall i \in \{0, ..., n-1\}$, $(i, a, i+1)$,
  - $\forall i \in \{1, ..., n\}$, $(i, \bar{a}, i-1)$,
- 0 est l´état initial de $\mathcal{A}_n$,
- $F_n = \{0, ..., n\}$ est l'ensemble des états finaux.

On note $\mathcal{L}_n$ le langage reconnu par l'automate $\mathcal{A}_n$. Dans la suite, ϵ désigne le mot vide.

1. Soient m, n des entiers naturels non nuls. Justifier précisément l'équivalence :

$$\mathcal{L}_m \subset \mathcal{L}_n \Leftrightarrow m \leqslant n.$$

   On pourra démontrer que $a^m \in \mathcal{L}_n$ équivaut à $m \leqslant n$.

2. Représenter l'automate $\mathcal{A}_1$; donner une expression rationnelle de $\mathcal{L}_1$.

3. Représenter l'automate $\mathcal{A}_2$; démontrer qu'une expression rationnelle de $\mathcal{L}_2$ est

$$\{\epsilon\} \cup a\{a\bar{a}, \bar{a}a\}^*\{\epsilon, a, \bar{a}\}.$$

4. Soit n un entier naturel non nul. Le but de ces questions est de déterminer une relation de récurrence qui permettrait de calculer une expression rationnelle de $\mathcal{L}_n$. Pour j dans $\{0, ..., n\}$, on note $\mathcal{L}_n^{0,j}$ le langage reconnu par l'automate $\mathcal{A}_n^{0,j} = (Q_n, \Delta_n, 0, \{j\})$, automate dont l'ensemble des états et l'ensemble des transitions sont identiques à ceux de $\mathcal{A}_n$ mais ayant le sommet j comme unique état final.

   (a) Démontrer la relation de récurrence : $\mathcal{L}_n^{0,0} = \{\epsilon\} \cup a\{\bar{a}a, \mathcal{L}_{n-1}^{0,0}\}^* \bar{a}$

   (b) Soit $j \in \{1, ..., n\}$. Démontrer : $\mathcal{L}_n^{0,j} = \mathcal{L}_n^{0,0} a \mathcal{L}_{n-1}^{0,j-1}$.

   (c) Conclure.

Exercice 6

Une compagnie d'avions dessert quatre villes : Angoulême, Bordeaux, Carcassonne, et Dax. Chaque avion ce cette compagnie respecte les règles suivantes :

(1) S'il dessert Dax, il fait escale à Bordeaux,

(2) S'il ne s'arrête pas à Bordeaux, il ne s'arrête pas à Angoulême.

(3) S'il dessert Carcassone, il fait escale à Angoulême.

(4) S'il ne s'arrête pas à Angoulême, il dessert Dax.

On introduit pour chacune de ces villes une variable propositionnelle: Pour Angoulême (respectivement Bordeaux, Carcassonne, Dax) , a (respectivement b, c, d) dont la valeur est `VRAI` si l'avion effectue un arrêt à Angoulême (respectivement Bordeaux, Carcassonne, Dax) et `FAUX` sinon.

1. Exprimer sous forme de propositions logiques, les quatre assertions (1), (2), (3) et (4). ces propositions seront notées respectivement $P_{(1)}, P_{(2)}, P_{(3)}, P_{(4)}$.

2. Dresser la table de vérité de chacune des propositions $P_{(i)}$, $i \in \{1, 2, 3, 4\}$, en fonction de celles de ces variables.

3. Démontrer :
$$(a \vee b \vee c \vee d) \wedge (P_{(1)} \wedge P_{(2)} \wedge P_{(3)} \wedge P_{(4)}) \Rightarrow b.$$
Que signifie ceci pour les avions de cette compagnie ?