F25Q

e3a

CONCOURS ENSAM - ESTP - EUCLIDE - ARCHIMEDE

Epreuve d'Informatique MP

durée 3 heures

L'usage de la calculatrice n'est pas autorisé

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

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

Exercice 1

a) Ecrire la procédure

`indicesMax`	`donnée`	T : `tableau d'entiers`
		l : `entier`
	`résultat`	m : `entier`
		n : `entier`

qui retourne le plus petit et le plus grand des indices correspondant à la valeur maximale d'un tableau de longueur l.

b) Ecrire la fonction

$$\texttt{moyenneNotesMinorées} \quad \texttt{données} \quad L : \texttt{liste d'entiers}$$
$$a : \texttt{entier}$$
$$\texttt{résultat} \quad x : \texttt{réel}$$

qui retourne la moyenne des notes supérieures ou égales à l'entier a dans une liste de notes. Cette liste contient des notes, qui sont des nombres compris entre 0 et 20, et se termine par la valeur -1. Dans le cas où cette liste ne comporte aucune note supérieure ou égale à l'entier a, la fonction `moyenneNotesMinorées` retournera la valeur -1 et affichera un message d'avertissement.

Exercice 2

a) Que calcule le programme suivant :

$u \leftarrow 0$
$v \leftarrow 50$
`tant que` $v > 0$ `faire`
    $u \leftarrow u + v^2$
    $v \leftarrow v - 1$
`fin tant que`
`afficher`(u)

b) Que calculent les fonctions suivantes :

$\texttt{A}(n : \textit{entier})$
    $\texttt{si}\,(n \geq 0)$ $\texttt{faire A}(n) \leftarrow n$
    $\texttt{sinon}$ $\texttt{faire A}(n) \leftarrow -n$

$\texttt{P}(n : \textit{entier})$
    $n \leftarrow \texttt{A}(n)$
    $\texttt{si}\,(n = 0)$ $\texttt{faire P}(n) \leftarrow 0$
    `sinon,` $\texttt{si}\,(n = 1)$ $\texttt{faire P}(n) \leftarrow 1$
        `sinon` $\texttt{faire P}(n) \leftarrow \texttt{P}(n-2)$

$\texttt{S}(m : \textit{entier}, n : \textit{entier})$
    $\texttt{si}\,(\texttt{A}(m) \leq \texttt{A}(n))$ $\texttt{faire S}(m,n) \leftarrow \texttt{A}(m)$
    `sinon` $\texttt{faire S}(m,n) \leftarrow \texttt{A}(n)$

$\mathtt{D}(m : entier, n : entier)$

$\mathtt{si}\,(m = 0)$ $\mathtt{faire}\ \mathtt{D}(m,n) \leftarrow \mathtt{A}(n)$
$\mathtt{si}\,(n = 0)$ $\mathtt{faire}\ \mathtt{D}(m,n) \leftarrow \mathtt{A}(m)$

$\mathtt{si}\,(m \neq 0)\,\mathtt{et\ si}\,(n \neq 0)$ $\mathtt{faire}$

$\mathtt{si}\,(\mathtt{P}(m) = 0)\,\mathtt{et}\,(\mathtt{P}(n) = 0)$ $\mathtt{faire}\ \mathtt{D}(m,n) \leftarrow 2 * \mathtt{D}(m/2, n/2)$
$\mathtt{si}\,(\mathtt{P}(m) = 0)\,\mathtt{et}\,(\mathtt{P}(n) = 1)$ $\mathtt{faire}\ \mathtt{D}(m,n) \leftarrow \mathtt{D}(m/2, n)$
$\mathtt{si}\,(\mathtt{P}(m) = 1)\,\mathtt{et}\,(\mathtt{P}(n) = 0)$ $\mathtt{faire}\ \mathtt{D}(m,n) \leftarrow \mathtt{D}(m, n/2)$
$\mathtt{si}\,(\mathtt{P}(m) = 1)\,\mathtt{et}\,(\mathtt{P}(n) = 1)$ $\mathtt{faire}\ \mathtt{D}(m,n) \leftarrow \mathtt{D}(\mathtt{A}(n) - \mathtt{A}(m), \mathtt{S}(m,n))$

Exercice 3

Soit n un entier naturel. On dit que c'est un 2-palindrome si son écriture en base 2 est la même qu'elle soit écrite de gauche à droite ou de droite à gauche. Plus précisément, si $n = \sum_{i=0}^{k} a_i 2^i$, avec $a_0, ..., a_k$ dans $\{0, 1\}$ et $a_k = 1$, n est un 2-palindrome si $a_i = a_{k-i}$, pour tout entier i dans $\{0, ..., k\}$. Par exemple, les entiers 3 (11), 5 (101), 7 (111), 9 (1001), et 15 (1111) sont des 2-palindromes. Ecrire un programme qui calcule les 2-palindromes n tels que $n \leq 511$.

Exercice 4

On considère l'alphabet à deux lettres: $\Sigma = \{a, b\}$.
Dans toute la suite, on fixe ω un mot non vide sur l'alphabet Σ ; on note n la longueur de ω (c'est à dire son nombre de lettres) et $a_1, ..., a_n$ ses n lettres lues de gauche à droite, c'est à dire que $\omega = a_1 a_2 ... a_n$.
Soit $\mathcal{A}_\omega = (Q, \Delta, 1, \{n + 1\})$ un automate sur l'alphabet Σ défini par :

- Q, l'ensemble des états de $\mathcal{A}_\omega$, est l'ensemble $\{1, 2, ..., n, n + 1\}$,
- Δ, l'ensemble des transitions de $\mathcal{A}_\omega$ ($\Delta \subset Q \times \Sigma \times Q$), est l'ensemble : $\{(1, a, 1), (1, b, 1), (i, a_i, i + 1), \forall i \in \{1, ..., n\}\}$;
- 1 est l´état initial de $\mathcal{A}_\omega$,
- $n + 1$ est l'unique état final de $\mathcal{A}_\omega$.

On note $\mathcal{L}(\mathcal{A}_\omega)$ le langage reconnu par l'automate $\mathcal{A}_\omega$.

1. On suppose que ω est la lettre a. Donc $n = 1$.

(a) Représenter l'automate $\mathcal{A}_a$ et justifier que $\mathcal{L}(\mathcal{A}_a)$ est le langage des mots se terminant par un a.

(b) Donner une expression rationnelle de $\mathcal{L}(\mathcal{A}_a)$.

(c) On considère l'automate $\mathcal{B}$ représenté sur la figure 1.

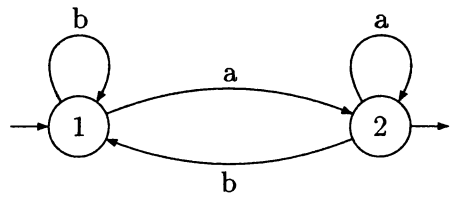

Figure 1: Automate $\mathcal{B}$

i. Démontrer que l'automate $\mathcal{B}$ est déterministe.

ii. Démontrer que le langage reconnu par l'automate $\mathcal{B}$ est égal à $\mathcal{L}(\mathcal{A}_a)$.

2. On suppose ici que $n > 1$ et que les lettres de ω sont toutes égales à a ($\forall i \in \{1, ..., n\}, a_i = a$).

(a) Représenter l'automate $\mathcal{A}_\omega$ et décrire le langage $\mathcal{L}(\mathcal{A}_\omega)$ dans ce cas.

On se propose de déterminiser l'automate $\mathcal{A}_\omega$: On applique à $\mathcal{A}_\omega$ l'algorithme de déterminisation et on note $\mathcal{C}_\omega$ l'automate ainsi obtenu.

(b) Montrer que les états de $\mathcal{C}_\omega$ sont :
$\{1\}, \{1,2\}, \{1,2,3\}, ..., \{1,2,...,i-1,i\}, ..., \{1,2,...,n\}\{1,2,...,n,n+1\}$,
et que les transitions de $\mathcal{C}_\omega$ sont alors :

- $(\{1\}, b, \{1\})$,
- $(\{1\}, a, \{1,2\})$,
- $(\{1,2,...,i-1,i\}, b, \{1\})$, pour tout i dans $\{2, ..., n+1\}$,
- $(\{1,2,...,i-1,i\}, a, (\{1,2,...,i,i+1\})$, pour tout i dans $\{2, ..., n+1\}$,
- $(\{1,2,...,n,n+1\}, a, (\{1,2,...,n,n+1\})$.

(c) Quel est l'état initial de $\mathcal{C}_\omega$?

(d) Quels sont les états finaux de $\mathcal{C}_\omega$?

(e) Représenter $\mathcal{C}_\omega$.

3. On suppose que n est pair et que $a_i = a$ lorsque i est impair, $a_i = b$ lorsque i est pair, c'est à dire que $\omega = abab...ab$. Construire un automate déterministe qui reconnait l'ensemble des mots finis sur l'alphabet Σ qui se terminent par ω.

Exercice 5

1. Soient g_1, g_2, g_3 trois fonctions booléennes. Justifier l'égalité : $g_1 + g_2g_3 = (g_1 + g_2)(g_1 + g_3)$.

2. Une fonction booléenne est dite *affine* si elle se décompose comme une somme de variables. Par exemple, $x+y+u$ et $x+v$ sont des fonctions affines des variables x, y, u, v. On considère la fonction booléenne f des 5 variables u, v, x, y, z, définie par :

$$f(u, v, x, y, z) = uv + vxy + vxz + uyz.$$

   Décomposer f comme un produit de fonctions booléennes affines (on pourra utiliser la question 1.).

3. Le directeur d'une banque souhaite, qu'en son absence, certains de ses collaborateurs, regroupés par deux ou trois selon un protocole établi précisément, puissent ouvrir le coffre. Pour cela, la porte du coffre est munie de n serrures et les n clés correspondantes sont nécessaires à l'ouverture de la porte. Le directeur dispose des clés des serrures en nombre suffisant et il en distribue certaines à ses collaborateurs de façon à ce que les seules possibilités soient :

   - Ulysse et Victoire peuvent à eux deux ouvrir le coffre
   - Victoire, Xavier et Yves peuvent à eux trois ouvrir le coffre,
   - Victoire, Xavier et Zoé peuvent à eux trois ouvrir le coffre,
   - Ulysse,Yves et Zoé peuvent à eux trois ouvrir le coffre.

   Déterminer le nombre minimal de serrures du coffre et une distribution convenable des clés.