CONCOURS ENSAM - ESTP - ARCHIMEDE

Épreuve d'Informatique MP

Durée 3 h

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 calculatrices est interdit.

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

Quel que soit le langage utilisé, le candidat pourra supposer qu'il dispose d'une fonction `test_liste_vide` qui prend en entrée une liste et renvoie le booléen 1 si la liste est vide et 0 sinon.

Le candidat pourra, s'il le juge utile, supposer que chaque liste se termine par un élément de marquage de fin de liste appelé NIL.

Exercice 1

a) Ecrire la fonction :

Tri_denombrement

	données	N : entier naturel
		L : liste d'entiers tous compris entre 1 et N
	résultat	M : tableau de longueur N

---

qui prend en entrée un entier naturel non nul N et une liste L d'entiers tous compris entre 1 et N et renvoie le tableau de longueur N, dans lequel la k-ième case contient le nombre d'éléments égaux à k dans la liste L, pour tout entier k compris entre 1 et N.

b) Calculer en fonction de n, n étant la longueur de la liste L, et de N, le coût de ce tri dans le meilleur des cas, dans le pire des cas et en moyenne. On supposera que le temps d'accès à la k-ième case du tableau est linéaire en k, le reste des opérations ayant un coût négligeable.

Exercice 2

(les questions a et b sont indépendantes.)

a) Ecrire la fonction :

Matrice

	données	L : liste
		n : entier
	résultat	M : tableau à deux entrées de taille n, n

---

qui prend en entrée une liste L et l'entier n et renvoie le tableau des coefficients d'une matrice carrée de taille n remplie ligne par ligne de gauche à droite par les coefficients de la liste L. La liste contient des entiers relatifs. Si la liste L n'est pas suffisamment longue, la matrice est complétée par des 0. Si la liste L est trop longue, le programme s'arrête lorsque la matrice est remplie.

b) Soit T un tableau rempli avec des entiers naturels. Un *plateau* de T de valeur k et de longueur l est une suite de l indices consécutifs $i, i+1, ..., i+l-1$ sur lesquels T prend la valeur k ($T[i] = T[i+1] = \cdots = T[i+l-1] = k$). Par exemple, le tableau $[1, 2, 3, 3, 3, 1, 2, 6, 6]$ posséde un plateau de valeur 3 et de longueur 3, un plateau de valeur 6 et de longueur 2, deux plateaux de valeur 1 et de longueur 1 et deux plateaux de valeur 2 et de longueur 1.
Ecrire la fonction :

LongueurMaxPlateau

	données	T : tableau
		N : entier naturel
	résultat	l : entier naturel

---

qui prend en entrée un tableau d'entiers naturels et sa longueur N et renvoie la plus grande longueur de ses plateaux.

Exercice 3

(les questions a, b et c sont indépendantes.)

a) Qu'effectue le programme ci-dessous :

```
PROG(L : liste d'entiers )
L2 ← liste_vide
E ← premier_element(L)
tant que (E <> NIL) faire
        E2 ← premier_element(L2)
        SW ← 0
        tant que(E2 <> NIL et SW = 0) faire
                si(E = E2) alors SW ← 1
                sinon E2 ← element_suivant(L2)
                fin si
        fin faire
        si (SW = 0) alors L2 ← ajouter_element(E)
        fin si
        E ← element_suivant(L)
fin faire
retournerL2
```

On remarquera que les listes considérées par ce programme se terminent par un élément de marquage de fin de liste appelé NIL.

b) Décrire l'exécution du programme NBS ci-dessous sur l'entrée $(7, 3)$:

```
NBS(N : entier strictement positif, p : entier strictement positif))
si (N < p) alors S ← 0
sinon si (p = 1) alors S ← 1
sinon S ← NBS(N − 1, p − 1) + NBS(N − p, p)
fin si
retourner(S)
```

Que calcule ce programme ?

c) Que calcule la procédure suivante :

$MS(N$: entier, T : tableau d'entiers de longueur N,)
  $U \leftarrow -1$
  $m \leftarrow -1$
  $n \leftarrow -1$
  $S \leftarrow -1$
  $b \leftarrow 0$
`pour` k `de` 1 `à` N `faire`
  `si` $(b = 0)$`et` $(T[k] \geqslant 0)$ `alors faire`
    $b \leftarrow 1$
    $i \leftarrow k$
    $j \leftarrow i$
    $S \leftarrow T[k]$
  `fin faire`
  `si` $(b = 1)$`et` $(T[k] \geqslant 0)$ `alors faire`
    $j \leftarrow k$
    $S \leftarrow S + T[k]$
  `fin faire`
  `si` $(T[k] < 0)$ `alors` $b \leftarrow 0$
  `fin si`
  `si` $(k = N)$ `alors` $b \leftarrow 0$
  `fin si`
  `si` $(b = 0)$`et` $(S > U)$ `alors faire`
    $U \leftarrow S$
    $m \leftarrow i$
    $n \leftarrow j$
  `fin faire`
  `fin si`
`fin faire`
`retourner` (U, m, n)

Exercice 4

Un nombre d'Armstrong est un nombre qui est égal à la somme des cubes des chiffres de son écriture en base 10 ; par exemple 153 est un nombre d'Armstrong puisque $153 = 1^3 + 5^3 + 3^3$. Ecrire un programme qui calcule tous les nombres d'Armstrong à trois ou quatre chiffres.

Exercice 5

Une liste d'entiers est dite *convenable* si elle se compose d'un nombre pair d'éléments $a_1, b_1, a_2, b_2, ...,$ a_k, b_k tels que $a_1 \leqslant b_1 < a_2 \leqslant \cdots b_{k-1} < a_k \leqslant b_k$. On représente une réunion finie de k intervalles fermés disjoints dont les extrémités sont des entiers relatifs par la liste ordonnée de leurs extrémités, selon l'ordre croissant. On admet que cette liste est alors convenable. Par exemple, $[-1,2] \cup [4,7] \cup [-3,-2] \cup [8,8]$ est représentée par la liste $-3,-2,-1,2,4,7,8,8$. L'ensemble vide est représenté par la liste vide.

a) Ecrire la procédure :

Convenable

`données`	L : `liste d'entiers`
`résultat`	b : `booléen`

qui prend en entrée une liste d'entiers L et retourne la valeur 1 si elle est convenable et 0 sinon.

b) On admet que l'intersection de deux réunions finies d'intervalles fermés disjoints dont les extrémités sont des entiers relatifs est aussi une réunion finie d'intervalles fermés disjoints dont les extrémités sont des entiers relatifs. Par exemple, l'intersection de $[1,2] \cup [4,5]$, représenté par la liste convenable $1,2,4,5$ et de $[-1,3] \cup [5,7] \cup [8,9]$, représenté par la liste convenable $-1,3,5,7,8,9$, est $[1,2] \cup [5,5]$ représenté par la liste convenable $1,2,5,5$. Ecrire la procédure :

Intersection

`données`	L_1 : `liste convenable d'entiers`
	L_2 : `liste convenable d'entiers`
`résultat`	L' : `liste convenable d'entiers`

qui prend en entrée deux listes convenables d'entiers représentant chacune une réunion finie d'intervalles fermés disjoints, respectivement F_1 et F_2, et retourne la liste convenable d'entiers représentant la décomposition en réunion finie d'intervalles fermés disjoints de l'ensemble $F_1 \cap F_2$.

c) On admet que la réunion de deux réunions finies d'intervalles fermés disjoints dont les extrémités sont des entiers relatifs est aussi une réunion finie d'intervalles fermés disjoints dont les extrémités sont des entiers relatifs. Par exemple, la réunion de $[1,2] \cup [4,5]$, représenté par la liste convenable $1,2,4,5$ et de $[-1,3] \cup [5,7] \cup [8,9]$, représenté par la liste convenable $-1,3,5,7,8,9$, est $[-1,3] \cup [4,7] \cup [8,9]$ représenté par la liste convenable $-1,3,4,7,8,9$. Ecrire la procédure :

Reunion

`données`	L_1 : `liste convenable d'entiers`
	L_2 : `liste convenable d'entiers`
`résultat`	L' : `liste convenable d'entiers`

qui prend en entrée deux listes convenables d'entiers représentant chacune une réunion finie d'intervalles fermés disjoints, respectivement F_1 et F_2, et retourne la liste convenable d'entiers représentant la décomposition en réunion finie d'intervalles fermés disjoints de l'ensemble $F_1 \cup F_2$.

Exercice 6

Si $\mathcal{L}$ est un langage sur un alphabet fini Σ et si n est un entier naturel non nul, on note $\mathcal{L}(n)$ le langage formé par les mots de $\mathcal{L}$ qui sont de longueur n.

1. Justifier que le langage $\mathcal{L}(n)$ est de cardinal fini.

Dans la suite, on note $u_n^{\mathcal{L}}$ le cardinal du langage $\mathcal{L}_n$.

2. Dans cette question, l'alphabet Σ désigne l'alphabet à trois lettres $\Sigma = \{a, b, c\}$ et $\mathcal{L}$ est le langage des mots finis sur Σ qui contiennent au moins un c. Ce langage est donné par l'expression rationnelle :
$$\mathcal{L} = \{a, b\}^* c \{a, b, c\}^*.$$
   a) Dessiner un automate déterministe à deux états qui reconnait exactement le langage $\mathcal{L}$.
   b) Soit n un entier naturel non nul.
      i) Dessiner un automate qui reconnait exactement le langage $\mathcal{L}(n)$.
      ii) On suppose $n \geqslant 2$. Soit $\mathcal{U}_{n-1}$ le langage $\{a, b, c\}^*(n-1)$. Démontrer que $\mathcal{L}(n)$ est la réunion disjointe des langages $\{a, b\}\mathcal{L}(n-1)$ et $c\,\mathcal{U}_{n-1}$. En déduire la relation de récurrence :
$$u_n^{\mathcal{L}} = 2u_{n-1}^{\mathcal{L}} + 3^{n-1}.$$
      iii) Exprimer $u_n^{\mathcal{L}}$ en fonction de n.

3. Dans cette question, on suppose que $\mathcal{L}$ est le langage reconnu par un automate déterministe $\mathcal{A} = (Q, \Sigma, 1, f, \delta)$ où :
   - Q est l'ensemble fini des états de $\mathcal{A}$. On note m son cardinal et on suppose que m est un entier naturel $\geqslant 2$. Les états dans Q sont numérotés de 1 à m.
   - 1 est le numéro de l'état initial de $\mathcal{A}$,
   - f est le numéro de l'unique état final de $\mathcal{A}$,
   - δ est la fonction de transition de $\mathcal{A}$ définie d'une partie de $Q \times \Sigma$ dans Q.

   On considère la matrice (m, m) $M = (m_{i,j})$ définie par : $m_{i,j}$ est le nombre de transitions de source l'état i et de but l'état j dans l'automate $\mathcal{A}$. On note $\chi_M(X) = X^m - \sum_{k=0}^{m-1} a_{m-k} X^k$ le polynôme caractéristique de la matrice M.

   Pour tout état j de Q, on note $\mathcal{L}_j$ le langage reconnu par l'automate $(Q, \Sigma, j, f, \delta)$; ainsi défini, $\mathcal{L}_1 = \mathcal{L}$. Pour tout entier $n \geqslant 1$, soit $V(n)$ le vecteur de coordonnées $x_1, ..., x_m$ où x_j est le cardinal du langage $\mathcal{L}_j(n)$, pour tout j dans Q.

   a) Expliciter les coordonnées du vecteur $V(1)$ en fonction de δ.
   b) Démontrer que, pour tout entier $n \geqslant 2$ et pour tout j dans Q, $\mathcal{L}_j(n)$ est la réunion disjointe des langages $a\mathcal{L}_k(n-1)$ pour tout (j, a, k) dans $Q \times \Sigma \times Q$ tel que $\delta(j, a) = k$.
   c) En déduire, pour tout entier $n \geqslant 2$, l'égalité $V(n) = M^{n-1}V(1)$.
   d) En utilisant le théorème de Cayley-Hamilton, démontrer que la suite $(u_n^{\mathcal{L}})$ vérifie la relation de récurrence :
$$u_{n+m}^{\mathcal{L}} = a_1 u_{n+m-1}^{\mathcal{L}} + a_2 u_{n+m-2}^{\mathcal{L}} + \cdots + a_m u_n^{\mathcal{L}}.$$
      Dans le cas où χ_M a m racines distinctes $\alpha_1, ..., \alpha_m$, que peut-on en déduire pour la suite $(u_n^{\mathcal{L}})$?

4. Soit $\mathcal{L}$ le langage sur l'alphabet $\Sigma = \{a, b\}$ reconnu par l'automate $\mathcal{B} = (Q, \Sigma, 1, 2, \delta)$ ci-dessous :

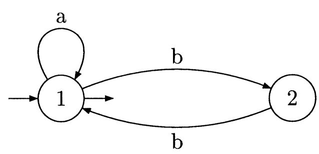

Figure 1: Automate $\mathcal{B}$

a) Donner une expression rationnelle de $\mathcal{L}$.
b) Expliciter la suite $(u_n^{\mathcal{L}})_{n\geqslant 1}$.

Exercice 7

Soit n un entier naturel $\geqslant 2$. Soit f une fonction booléenne à n variables. On dit que f définit implicitement sa n-ième variable si pour tout n-uplet $(x_1, ..., x_{n-1})$ de $\{0,1\}^{n-1}$, il existe un unique x_n dans $\{0,1\}$ tel que $f(x_1, ..., x_n) = 0$.

1. Dans cette question, $n = 2$. Soit f_1 la fonction boolénne définie par ses valeurs présentées dans la table ci-dessous :

x_1	x_2	$f_1(x_1, x_2)$
0	0	1
0	1	0
1	0	0
1	1	1

Démontrer que f_1 définit implicitement sa seconde variable. Expliciter toutes les fonctions booléennes sur $\{0,1\}^2$ qui définissent implicitement leur seconde variable.

Soit f une fonction booléenne à n variables.

2. Démontrer que f définit implicitement sa n-ième variable si et seulement si f vérifie la propriété suivante :
$$\forall (x_1, ..., x_{n-1}, x_n) \in \{0,1\}^n, f(x_1, ..., x_{n-1}, \overline{x}_n) = \overline{f(x_1, ..., x_n)}.$$

3. Soit f la fonction booléenne définie par :
$$\forall (x_1, ..., x_{n-1}, x_n) \in \{0,1\}^n, f(x_1, ..., x_n) = x_1 x_2 ... x_n + \sum_{i=1}^{n-1} \overline{x}_i \overline{x}_n.$$
Démontrer que f définit implicitement sa n-ième variable. Démontrer plus précisément qu'il existe une fonction booléenne g sur $\{0,1\}^{n-1}$, telle que
$$\forall (x_1, ..., x_{n-1}, x_n) \in \{0,1\}^n, \ (\ f(x_1, ..., x_n) = 0 \Leftrightarrow x_n = g(x_1, ..., x_{n-1})\ ).$$
Préciser g.