

Modalités :

- Durée : 2 heures
- Aucun document autorisé, ni machine à calculer, ni téléphone
- Le barème est donné à titre indicatif (± 1)
- **Une feuille par exercice**

Exercice 1 (7 points) – Dictionnaire.

On souhaite construire un arbre binaire de recherche nommé *Dictionnaire* dont les nœuds contiennent des couples de la forme suivante (un mot, longueur de ce mot). Un mot est une chaîne de caractères.

- 1) Étant donné un *Dictionnaire* d et un *Mot* m , écrire l'algorithme récurif du sous-programme **insérerMot** qui insère le mot m à la bonne place dans le *Dictionnaire* d . (3 points)
On suppose que l'on dispose des deux fonctions utilitaires suivantes :
 - *longueur*($m : \text{Mot}$) : *Entier*, qui retourne la longueur du mot m .
 - *lexico*($m_1 : \text{Mot}, m_2 : \text{Mot}$) : *Entier*, qui retourne
 - 1 si $m_1 > m_2$
 - 0 si $m_1 = m_2$
 - -1 si $m_1 < m_2$
- 2) Donner en langage C la déclaration complète du type *Dictionnaire*. (1 point)
- 3) Étant donnée une liste l de mots, écrire l'algorithme récurif du sous-programme **construireDico** qui construit le *Dictionnaire* contenant tous les mots de la liste l et leur longueur respective. (3 points)

Exercice 2 (10 points) – Listes.

On souhaite analyser la pluviométrie sur un mois. On considère une pluviométrie mensuelle comme la liste des relevés pluviométriques journaliers du mois considéré. Un relevé pluviométrique journalier est caractérisé par :

- 1) un jour (entier strictement positif compris entre 1 et maximum 31),
- 2) la hauteur d'eau en millimètres tombée ce jour-là (entier strictement supérieur à 0).

Attention :
- ne figurent dans la liste que les jours du mois où il a plu,
- la liste est triée selon l'ordre croissant des jours.

Exemple :

Pour un mois donné, s'il a plu le 2 (12 mm), le 3 (10mm), le 4 (5mm), le 7 (3mm), le 8 (14mm), le 12 (5mm), le 20 (7mm), le 21 (7mm) et le 24 (1mm) et qu'il n'a pas plu les autres jours, alors on a la liste suivante :

(2, 12), (3, 10), (4, 5), (7, 3), (8, 14), (12, 5), (20, 7), (21, 7), (24, 1)

- 1) Étant données une liste de relevés pluviométriques et une hauteur h souhaitée, écrire l'algorithme récurif du sous-programme **plusHaut** permettant de construire une nouvelle liste contenant tous les jours où il a plu plus que la hauteur désirée. (2 points)
sur l'exemple donné et pour $h=11$ alors la liste à construire est : 2, 8
- 2) Étant donnée une liste de relevés pluviométriques, écrire l'algorithme du sous-programme itératif **pluvioMax** permettant de calculer la période (en nombre de jours consécutifs) la plus longue où il a plu tous les jours. (4 points)
3 sur l'exemple donné
- 3) Étant donnée une liste de n listes de relevés pluviométriques correspondantes à n mois consécutifs, écrire l'algorithme itératif **moyenneN** qui construit la liste contenant le relevé pluviométrique moyen par jour sur ces n mois en englobant les jours avec pluviométrie nulle. La liste doit donc contenir tous les jours de 1 à 31, et elle est triée par ordre croissant sur les jours. (4 points)

Exercice 3 (3 points) – Grammaire et Langage.

Soit la grammaire $G = (\{X, Y\}, \{a, b\}, \rightarrow, X)$ avec \rightarrow défini de la manière suivante:

1. $X \rightarrow aX \mid Y$
2. $Y \rightarrow bY \mid b$

- 1) Montrer que le mot $aaab$ appartient au langage généré par G . (1 point)
- 2) Quel est la forme générale (à exprimer à l'aide des deux lettres a et b ; une expression régulière de la forme des mots du langage) des mots appartenant à ce langage $L(G)$. (2 points)

Terms:

- Duration: 2 hours
- Documents, calculator and phone are not authorized
- The rate is indicative (± 1)
- **One sheet per exercise**

Exercise 1 (7 points) – Dictionary.

We want to build a binary search tree named *Dictionary* whose nodes contain pairs of the following form (a word, length of this word). A word is a string.

1) Given a *Dictionary* d and a *Word* m , write the recursive algorithm of the subroutine **insertWord** which inserts the word m at the right place in the *Dictionary* d . (3 points)

You can use the following two utilitarian functions :

- $length(m: Word) : Integer$, which returns the length of the word m .
- $lexico(m_1: Word, m_2: Word) : Integer$, which returns
 - 1 if $m_1 > m_2$
 - 0 if $m_1 = m_2$
 - -1 if $m_1 < m_2$

2) Give in C language the complete declaration of the type *Dictionary*. (1 points)

3) Given a list l of words, write the recursive algorithm of the subprogram **buildDictio** building the *Dictionary* containing all the words of the list l and their respective length. (3 points)

Exercise 2 (10 points) – Lists.

We want to analyze the rainfall over a month. Monthly rainfall is considered as a list of daily rainfall records for the considered month. A daily rainfall record is characterized by:

- 1) a day (strictly positive integer between 1 and 31),
- 2) the precipitation (height of water) in millimeters fell on that day (integer strictly greater than 0).

Be careful:

- Only the days of the month, in which it rained, are listed
- the list is sorted in ascending order of days.

Example: For a given month, the 2 (12 mm), 3 (10mm), 4 (5mm), 7 (3mm), 8 (14mm), 12 (5mm), the 7mm), 21 (7mm) and 24 (1mm) and it does not have the other days, then we have the following list:

(2,12), (3,10), (4,5), (7,3), (8,14), (12,5), (20,7), (21,7), (24,1)

1) Given a list of rainfall records and a desired height h , write the recursive algorithm of the subroutine **higher** to construct a new list containing every day where it has rained more than the specified height. (2 points)

In the above example and for $h = 11$ then the list to be constructed is: 2, 8

2) Given a list of rainfall records, write the iterative algorithm of the subroutine to compute the longest period (number of consecutive days) in which it rained every day. (4 points)

3 in the above example

3) Given a list of n lists of rainfall records corresponding to n consecutive months, write the iterative algorithm **averageN** that builds the list of average rainfall records containing the average rainfall per day over these n months, including the days with zero rainfall. The list must therefore contain every day from 1 to 31, and it is sorted in ascending order on the days. (4 points)

Exercise 3 (3 points) – Grammar and language.

Let the grammar $G = (\{X, Y\}, \{a, b\}, \rightarrow, X)$ with \rightarrow defined in the following way:

1. $X \rightarrow aX \mid Y$
2. $Y \rightarrow bY \mid B$

1) Show that the word *aaab* belongs to the language generated by G . (1 point)

2) What is the general form (to be expressed using the two letters a and b , a regular expression) of the words belonging to this language $L(G)$? (2 points)