

Soutien Algorithmique 2

(ancien sujet d'examen)

Morse

On souhaite traduire des textes écrits avec l'alphabet habituel pour obtenir des textes en morse, et *vice versa*. Dans ce problème, on ne s'intéresse qu'au passage d'un caractère de l'alphabet à un code morse (codage caractère-morse), et à l'opération inverse (décodage caractère-morse).

Un caractère morse est une séquence de tirets et de points. L'alphabet morse est défini par le tableau suivant :

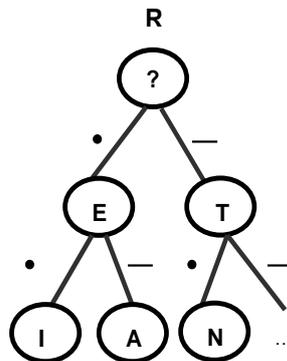
A	•-	F	•••	K	-•-	P	•--•	U	••-
B	-•••	G	--•	L	••••	Q	--•-	V	•••-
C	-•-•	H	••••	M	--	R	••	W	•--
D	-••	I	••	N	-•	S	•••	X	-••-
E	•	J	•---	O	---	T	-	Y	-•-
Z	--••								

On structure l'ensemble des codes de l'alphabet morse sous forme d'arbre binaire en mettant à profit le fait que ces codes sont construits à partir des deux caractères '-' et '•' :

Chaque nœud de l'arbre contient un caractère alphabétique, chaque lien correspond à un caractère du codage morse (point ou trait).

La racine de l'arbre correspond à un caractère fictif '?'.

Le chemin allant de la racine à un nœud correspond au code morse traduisant le caractère alphabétique de ce nœud :



Question 1

Dessinez l'arbre complet comprenant toutes les lettres de l'alphabet.

Question 2

On dispose des fonctions **gauche(x)**, **droit(x)**, **père(x)** et **info(x)** qui désignent respectivement le sous-arbre gauche, le sous-arbre droit, le nœud père et l'information associée à un nœud **x** donné, ainsi que les fonctions **existeGauche(x)**, **existeDroit(x)** et **estFeuille(x)** qui renvoient un booléen indiquant respectivement si **x** a un sous-arbre gauche, un sous-arbre droit, ou si **x** est une feuille.

a) Écrire à l'aide de ces fonctions la fonction (récursive ou non) **morseCar1** qui calcule le caractère alphabétique correspondant à une chaîne morse donnée :

fonction morseCar1 (m : chaîne, x : Arbre) → caractère

// MorseCar1(m,x) renvoie le caractère correspondant à la chaîne morse m décodée à partir du
// nœud x de l'arbre

Exemples : morseCar1("•-",r) = 'A'
morseCar1("•-••",r) = 'L'

r désigne la racine de l'arbre représentant le codage en morse.

b) Écrire à l'aide de mêmes fonctions la fonction (récursive ou non) **carMorse1** qui calcule la chaîne morse traduisant un caractère alphabétique donné :

fonction carMorse1(c : caractère, x : Arbre) → chaîne

// CarMorse1(c,x) renvoie la chaîne morse correspondant au caractère c
// recherché dans le sous-arbre de racine X

Exemples : carMorse1('A',R) = "•-"
carMorse1('L',R) = "•-••"

Question 3

On décide de représenter cet arbre comme un arbre binaire complet dans un tableau de caractères appelé **tmorse**. Lorsqu'un arbre binaire complet est représenté dans un tableau, alors les fils du nœud situé à l'indice N sont situés aux index 2^N et 2^{N+1} , s'ils existent). On complétera les positions manquantes de l'arbre donné en question 1) par des caractères fictifs, pour en faire un arbre binaire complet.

a) Donnez le contenu du tableau tmorse.

tmorse	?	E	T	...				
	1	2	3	4	5	...		

b) Écrivez les deux fonctions de codage d'un caractère vers le morse (carMorse), et de décodage d'une chaîne morse vers un caractère alphanumérique (morseCar) :

En remplaçant les appels aux fonctions par leur réalisation en termes de manipulation du tableau tmorse, par exemple : info(x) sera remplacé par tmorse[x], gauche(x) par x*2, etc.

fonction carMorse (x : caractère) → chaîne

// carMorse(X) désigne la chaîne de caractères morse codant le caractère X

Exemple : carMorse('A')="•-"
carMorse('4')="••••-"

fonction morseCar (x : chaîne) → caractère

// MorseCar(x) désigne le caractère alphanumérique codé par la chaîne de caractères morse X

Utiliser des fonctions intermédiaires carMorse2 et morseCar2 du même modèle que les fonctions spécifiées à la question 2, en les adaptant à la représentation définie pour représenter l'arbre.