

## Examen

### Exercice 1 LLCs (8.5 pts : 0.5 + 1.5 + 1.5 + 2 + 2 + 1)

Le code morse a  t  invent  par Samuel Morse, peintre et physicien Am ricain. C'est un code permettant de transmettre un texte   l'aide de s ries d'impulsions courtes et longues. Il a  t  longtemps utilis  pour effectuer des liaisons longue distance.

En informatique, et plus particuli rement dans les textes, une impulsion longue est repr sent e par un tiret '-' et une impulsion courte par un point '.'. Chaque caract re de l'alphabet est pr sent  par un code morse.

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

Exemple :

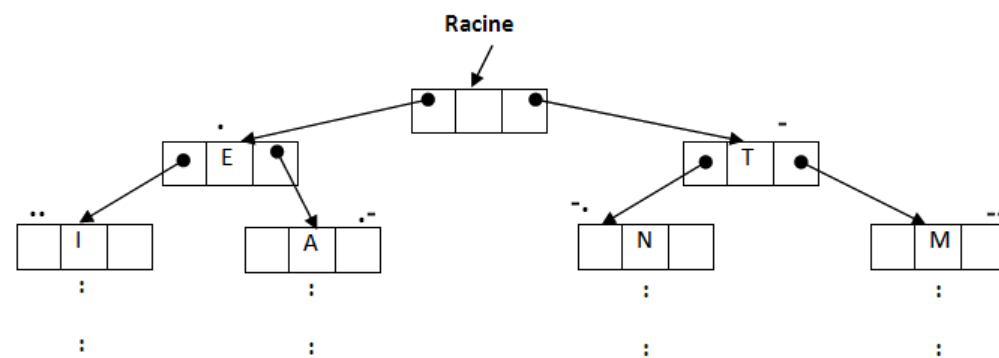


### Questions :

1. Donner les structures de donn es n cessaires   la repr sentation de la liste des codes morse.
2. Ecrire la proc dure AjouterCode(Tete, car, cde) permettant d'ajouter la lettre car avec son code cde   la liste des codes.
3. Ecrire la proc dure RetirerCar(Tete, car) permettant de supprimer le caract re car de la liste des codes.
4. Ecrire la fonction Coder(Tete, Ch) permettant de retourner le code morse de la chaine de caract res Ch. Les codes morse de deux caract res sont s par s par un espace, tandis que les codes morses de deux mots sont s par s par deux espaces (blancs).
5. Ecrire la fonction Decoder(Tete, ChMorse) permettant de retourner la chaine de caract res cod e dans la chaine morse ChMorse.
6. Calculer la complexit  des fonctions Coder et D coder.

### Exercice 2 ARB (9.5 pts : 0.5 + 3 + 2 + 3 + 1)

On souhaite repr senter la liste des codes morse pr sent e dans l'exercice pr c dent par un arbre de recherche binaire, o  l'orientation vers un fils gauche signifie un '.' et l'orientation vers un fils droit signifie un '-'.



### Questions :

1. Donner les structures de données nécessaires à la représentation de l'arbre des codes morses.
2. Ecrire la procédure AjouterCode(Racine, car, cde) permettant d'ajouter la lettre car avec son code cde à l'arbre des codes.
3. Ecrire la fonction Coder(Racine, Ch) permettant de retourner le code morse de la chaîne de caractères Ch. Les codes morse de deux caractères sont séparés par un espace, tandis que les codes morses de deux mots sont séparés par deux espaces (blancs).
4. Ecrire la fonction Décoder(Racine, ChMorse) permettant de retourner la chaîne de caractères codées par la chaîne morse ChMorse.
5. Calculer la complexité des fonctions Coder et Décoder. Que peut-on conclure par rapport à l'exercice précédent ?

**Exercice 3** Tas (2 pts : 1.5 + 0.5)

1. Donner le tas statique obtenu après l'insertion dans l'ordre des clés suivantes :

**12, 3, 5, 14, 15, 8, 1, 6**

2. Donner le tas après deux retraits

★★★ Bonne chance ★★★

Dr A.Djeffal

## Corrigé type

### Exercice 1 : Listes linéaires chaînées

1. Structures de données (0.5 pt)

```
Type TMaillon = Structure
    Car : caractère ;
    Code : chaîne[4] ;
Fin ;

Var Tete : Pointeur(TMaillon) ;
```

2. Procédure AjouterCode(Tete, Car, Cde) (1.5 pt)

```

Procédure AjouterCode( Tete : Pointer(TMaillon), Car :
caractère, Cde : Chaine[4]);
Var P : Pointeur(TMaillon);
D[U+FFFF]t
  P ← Tete;
  Tant que ((P ≠ Nil et Car(P) ≠ Car et Code(P) ≠
  Cde)faire
    | P ← Suivant(P);
  Fin TQ;
  Si (P ≠ Nil) Alors
    | Ecrire('Le code existe déjà')
  Sinon
    Allouer(P);
    Aff_Car(P, Car);
    Aff_Code(P, Cde);
    Aff_Adr(P, Tete);
    Tete ← P;
  Fin Si;
Fin;

```