

Mode T9

3h (avec documents)

Année 2024-2025

Préambule

- Pour tester dans utop : **open Be;** puis ouverture des modules associés aux fichiers dans lesquels il y a les fonctions que vous voulez tester.
- Le code rendu doit **impérativement compiler**. Si une partie ne compile pas, mettez-la en commentaires, cela peut donner lieu à des points.
- Interdiction de modifier les signatures (nom des fonctions + type) indiquées dans les contrats.
- Même si la non utilisation d'itérateur sera pénalisée, des points seront accordés si les fonctions sont écrites correctement sans itérateur.
- Les documents autorisés sont : vos notes de cours, TD et TP, le site d'OCaml et Moodle. Vous êtes autorisé (et incité !) à utiliser toutes les fonctions de la librairie standard, en particulier le module List : <https://ocaml.org/api/List.html>.
- L'utilisation de modèles de langage (ChatGPT, Copilot ...) et d'outils de partage de code est interdite.
- Utilisez le script `genererRendu.sh` pour générer un fichier `<votre_login>.tar` compression d'un répertoire `<votre_login>` contenant les fichiers `.mli` et `.ml`, le dictionnaire français et les fichiers dune. Avant de déposer le tar, vérifiez qu'il contient bien les fichiers souhaités.

Barème prévisionnel

Manipulation des booléens : 1 pt

Exercice	1	2	3	4	5
Points	8	8	2	2	1

Total : 22 pt

1 Envoi de SMS avant l'aire des smartphones

Les claviers des anciens modèles de téléphones portables (voir [1a](#)) sont dotés de touches qui comportent plusieurs lettres (3 ou 4) (voir [1b](#)). Pour envoyer des SMS, il faut appuyer (éventuellement plusieurs fois) sur les touches correspondants à chaque lettre du mot.

Sans saisie intuitive Pour saisir une lettre, il faut taper plusieurs fois sur une touche pour faire défiler les lettres correspondant à la touche. Par exemple, le **c** est situé sur la touche 2, la position du **c** est la troisième de **abc**, il faut donc appuyer 3 fois sur le 2 pour saisir un **c**. Donc pour écrire **bonjour** il faut appuyer :

- 2 fois sur 2
- 3 fois sur 6
- 2 fois sur 6
- 1 fois sur 5
- 3 fois sur 6
- 2 fois sur 8
- 3 fois sur 7

Au total, il faut appuyer **16 fois** sur les touches pour écrire ce mot.



(a) Téléphone à touches



(b) Zoom sur les touches

FIGURE 1 – Les téléphones avant les smartphones

Avec saisie intuitive : mode T9 Le mode T9 (text on 9 keys) facilite la saisie des textes. Dans le mode T9, une seule frappe par lettre suffit et le téléphone propose de lui-même les mots, à partir d'un dictionnaire, qui correspondent à la séquence de touches qui vient d'être tapée.

Par exemple, la suite de touches 2, 6, 6, 5, 6, 8 et 7 correspond au mot *bonjour*. Cependant, il est possible qu'une suite de touches corresponde à plusieurs mots. Ainsi, la suite 8, 3, 6, 3, 7, 3 correspond aux mots *tendre* et *vendre*.

2 Description des fichiers fournis

- `chaines.mli` et `chaines.ml` : identiques aux fichiers fournis pour le TP sur les arbres lexicographiques, pour décomposer et recomposer des chaînes de caractères.
- `encodage.ml` : définit une interface et deux modules associant à chaque touche du clavier numérique un ensemble de lettres. Deux configurations du clavier numérique sont proposées : celle correspondant au mode T9 et une où les voyelles sont associées à la touche 2 et les consonnes à la touche 3.
- `structureDonnees.ml` : A COMPLETER
Définit un module signature décrivant les structures de données possibles pour implanter les dictionnaires. Deux implantations devront être fournies : la première est basée sur les listes associatives et la seconde sur des arbres n-aires.
- `intuitive.ml` : A COMPLETER
Définit un foncteur paramétré par un encodage et une structure de données et permet de coder les fonctions nécessaires à la saisie intuitive.
- `testStupide.ml` et `testT9.ml`
Instancie le foncteur et teste la saisie intuitive dans les différentes configurations, c'est-à-dire avec les différents encodages et les différentes structures de données pour coder le dictionnaire.

3 Structure de données : liste associative

La première structure de données pour coder le dictionnaire est une liste associative. Une liste de mots (`string list`) est associées à une suite de touches (`int list`).

```
type dico = ( int list * string list ) list
```

Cette liste ne doit pas contenir de clés en double : autrement dit, pour chaque couple de la liste, les premiers éléments doivent être distincts entre eux.

▷ **Exercice 1** Dans le fichier `structureDonnees.ml`, définir un module `ListAssoc` compatible avec l'interface `StructureDonnees`.

Les tests des fonctions sont donnés dans le fichier `testListAssoc.txt` sauf ceux de la fonction `ajouter` qui sont à écrire.

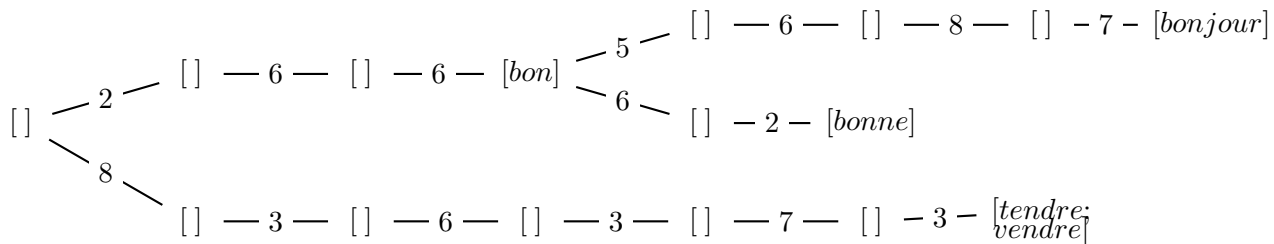
4 Structure de données : arbre n-aire

La structure de données de type liste associative ne convient pas à notre cas d'utilisation. Vous constaterez par la suite que certains tests effectués sur de grands dictionnaires sont très lents lorsqu'on utilise ces listes associatives.

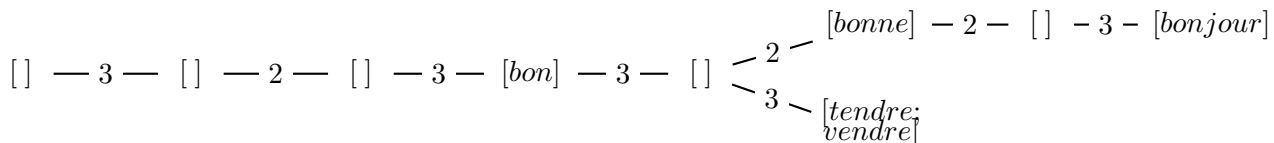
Nous proposons une autre structure de données pour le dictionnaire : elle est représentée par un arbre n-aire, où les nœuds contiennent des listes de mots et les branches sont marquées par des chiffres (touches).

```
type dico = Noeud of ( string list * ( int * dico ) list )
```

Pour le mode T9, le dictionnaire composé des mots "bon", "bonne", "bonjour", "vendre" et "tendre" est le suivant :



Pour le mode voyelles / consonnes, le dictionnaire composé des mêmes mots est le suivant :



Même si c'est le cas dans les exemples, il n'est pas demandé que les branches soient ordonnées.

▷ **Exercice 2** Dans le fichier `structureDonnees.ml`, définir un module `Arbre` compatible avec l'interface `StructureDonnees`.

Les tests des fonctions sont donnés dans le fichier `testArbre.txt`.

5 Saisie intuitive

Les exercices suivants sont à faire dans le fichier `intuitive.ml`

▷ Exercice 3 *Mots* → *Touches*

1. Écrire la fonction `encoder_lettre` qui indique la touche associée à la lettre passée en paramètre.
2. Écrire la fonction `encoder_mot` qui calcule la suite de touches à presser pour saisir un mot passé en paramètre.

▷ Exercice 4 *Touches* → *Mots*

Pour les tests, une liste de mots du langage français est fournie (Petit Larousse Illustré 2007). Une fonction permettant de construire un dictionnaire à partir du fichier est également fournie. Elle s'appuie sur `empty` et `ajouter`. Si une de ses fonctions est fausse, les tests utilisant `dico_fr` (encodage T9) et `dico_fr_stupide` (encodage voyelles / consonnes) échoueront.

1. Écrire la fonction `ajouter` qui ajoute un mot à un dictionnaire.
2. Écrire la fonction `decoder_mot` qui identifie l'ensemble des mots correspondant à une suite de touches dans un dictionnaire.
3. Écrire la fonction `max_mots_code_identique` qui calcule le nombre maximal de mots ayant la même séquence de touches dans un dictionnaire.
4. Écrire la fonction `prefixe` qui liste l'ensemble des mots d'un dictionnaire dont le préfixe a été saisi.

6 Tests du foncteur Intuitive

▷ Exercice 5 Dans les fichiers `testStupide.ml` et `testT9.ml`, instanciez le foncteur `Intuitive` de façon à lancer les 4 jeux de tests :

- Mode T9 et liste associative
- Mode T9 et structure arborescente
- Mode stupide et liste associative
- Mode stupide et structure arborescente

Attention, les tests utilisant les listes associatives sont longs sur le dictionnaire français.