

# Mode T9

3h (avec documents)

Année 2023-2024

## 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>.
- Les exercices sont indépendants mais au sein d'un même exercice, il y a des dépendances.
- Utilisez le script `genererRendu.sh` pour générer un fichier `<votre_login>.tar` compression d'un répertoire `<votre_login>` contenant les fichiers `.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 : 0.5 pt
- Exercice 2 : 5.5 pt
- Exercice 3 : 3.5 pt
- Exercice 4 : 2.5 pt
- Exercice 5 : 8.5 pt
- Total : 21.5 pt

## 1 Envoi de SMS avant l'air 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



(a) Téléphone à touches



(b) Zoom sur les touches

FIGURE 1 – Les téléphones avant les smartphones

- 2 fois sur 8
- 3 fois sur 7

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

**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` : définition d'un type 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.
- `naif.ml` : à compléter avec les fonctions correspondant à la saisie sans mode intuitif.
- `intuitive.ml` : à compléter avec les fonctions correspondant à la saisie intuitive.

## 3 Sans saisie intuitive

Comme indiqué précédemment, sans saisie intuitive, il faut appuyer plusieurs fois sur une touche pour saisir une lettre. Quand deux lettres successives correspondent à la même touche, c'est une pause dans la saisie qui indique le passage à la lettre suivante. Pour faciliter l'écriture des fonctions, nous remplaçons cette pause par un appui sur la touche 0. Ainsi pour saisir "bonjour", il faudra la suite de touches 2, 2, 0, 6, 6, 0, 6, 6, 0, 1, 0, 6, 6, 6, 0, 8, 8, 0, 7, 7, 7, 0.

▷ **Exercice 1** Définir deux exceptions `NotALetter` et `CodeInvalide`.

## ▷ Exercice 2 Mots → touches

1. Écrire la fonction `encoder_lettre` qui indique la touche et le nombre de fois qu'il faut appuyer dessus pour saisir la lettre passée en paramètre. La liste associative (touche, liste de lettres) est passée en paramètre de la fonction.
2. Écrire le contrat d'une fonction `creer_liste` qui créer une liste de taille  $[n]$  dont tous les éléments sont identiques à un élément `e`.
3. Écrire les tests unitaires de la fonction `creer_liste`.
4. Écrire la fonction `creer_liste`.
5. Écrire la fonction `encoder_mot` qui calcule la suite de touche à presser pour saisir un mot passé en paramètre. La liste associative (touche, liste de lettres) est passée en paramètre de la fonction.

## ▷ Exercice 3 Touches → mot

1. Écrire la fonction `decoder_lettre` qui identifie la lettre saisie à partir d'une touche et du nombre de fois qu'elle a été pressée. La liste associative (touche, liste de lettres) est passée en paramètre de la fonction.
2. Écrire la fonction `decoder_mot` qui identifie le mot saisi à partir d'une suite de touches. La liste associative (touche, liste de lettres) est passée en paramètre de la fonction.

# 4 Avec saisie intuitive

## ▷ Exercice 4 Mots → touches

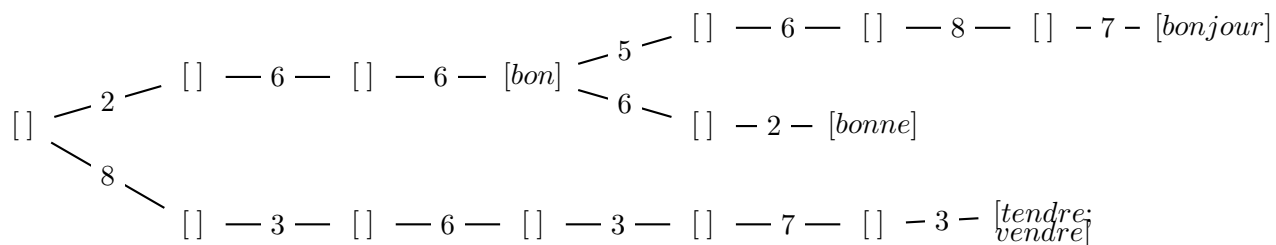
1. Écrire la fonction `encoder_lettre` qui indique la touche associée à la lettre passée en paramètre. La liste associative (touche, liste de lettres) est passée en paramètre de la fonction. La touche associée aux caractères spéciaux est la touche 0.
2. Écrire la fonction `encoder_mot` qui calcule la suite de touches à presser pour saisir un mot passé en paramètre. La liste associative (touche, liste de lettres) est passée en paramètre de la fonction.

**Dictionnaire** Comme indiqué précédemment, le mode de saisie intuitive repose sur un dictionnaire qui à une suite de touche associe un ensemble de mots.

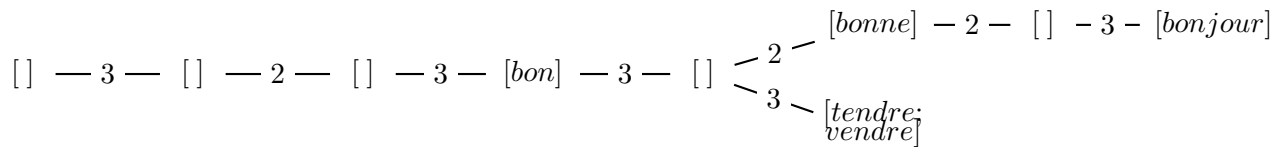
Ce dictionnaire est représenté par un arbre n-aire avec des listes de mots dans les nœuds et des chiffres (touches) sur les branches :

```
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.** 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. D’autres tests sont fournis avec des dictionnaires créés à la main.

▷ **Exercice 5 Touches → Mots**

1. Écrire la fonction `empty` qui créer un dictionnaire vide.
2. Écrire la fonction `ajouter` qui ajoute un mot à un dictionnaire. La liste associative (touche, liste de lettres) est passée en paramètre de la fonction.
3. Écrire la fonction `decoder_mot` qui identifie l’ensemble des mots correspondant à une suite de touches dans un dictionnaire.
4. Écrire la fonction `max_mots_code_identique` qui calcule le nombre maximal de mots ayant la même séquence de touches dans un dictionnaire.
5. Écrire la fonction `lister` qui liste l’ensemble des mots d’un dictionnaire.
6. Écrire la fonction `prefix` qui liste l’ensemble des mots d’un dictionnaire dont le préfixe a été saisi.