

# Le codage T9

Émilie ROGER et Alexia TARTAS

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
*	0 +	#



# Table des matières

<b>1</b>	<b>Notice utilisateur</b>	<b>4</b>
1.1	Lancer le programme	4
1.2	Naviguer entre les différentes fonctions : la page d'accueil	4
1.3	Utiliser la première fonction : tapez 1	5
1.4	Utiliser la deuxième fonction : tapez 2	6
1.5	Utiliser la troisième fonction : tapez 3	7
1.6	Utiliser la quatrième fonction : tapez 4	9
1.7	Remarques	11
<b>2</b>	<b>Structuration des données</b>	<b>12</b>
2.1	Architecture	12
2.2	Fonction « lectureDico »	13
2.2.1	Description	13
2.2.2	Fonctionnement	13
2.3	Fonction « Accueil »	13
2.3.1	Description	13
2.3.2	Fonctionnement	13
2.4	Fonction « codageMultiTap »	14
2.4.1	Description	14
2.4.2	Fonctionnement	14
2.5	Fonction « codageT9 »	14
2.5.1	Description	14
2.5.2	Fonctionnement	14
2.6	Fonction « decodageMultiTap »	15
2.6.1	Description	15
2.6.2	Fonctionnement	15

<b>2.7</b>	<b>Fonction « corresDico »</b>	<b>16</b>
2.7.1	Description	16
2.7.2	Fonctionnement	16
<b>2.8</b>	<b>Procédure « decodageT9 »</b>	<b>17</b>
2.8.1	Description	17
2.8.2	Fonctionnement	17
<b>3</b>	<b>Exemples</b>	<b>19</b>
<b>3.1</b>	<b>Fonction « Accueil »</b>	<b>19</b>
<b>3.2</b>	<b>Fonction « codageMultiTap »</b>	<b>19</b>
<b>3.3</b>	<b>Fonction « codageT9 »</b>	<b>20</b>
<b>3.4</b>	<b>Fonction « decodageMultiTap »</b>	<b>20</b>
<b>3.5</b>	<b>Procédure « decodageT9 »</b>	<b>20</b>
<b>4</b>	<b>Idées et évolutions</b>	<b>21</b>
<b>4.1</b>	<b>Idées</b>	<b>21</b>
<b>4.2</b>	<b>Évolutions</b>	<b>22</b>
4.2.1	Pour la première et la deuxième fonction	22
4.2.2	Pour la troisième et la quatrième fonction	22
4.2.3	Pour la fonction decodageMultiTap	22
4.2.4	Pour la procédure decodageT9	22

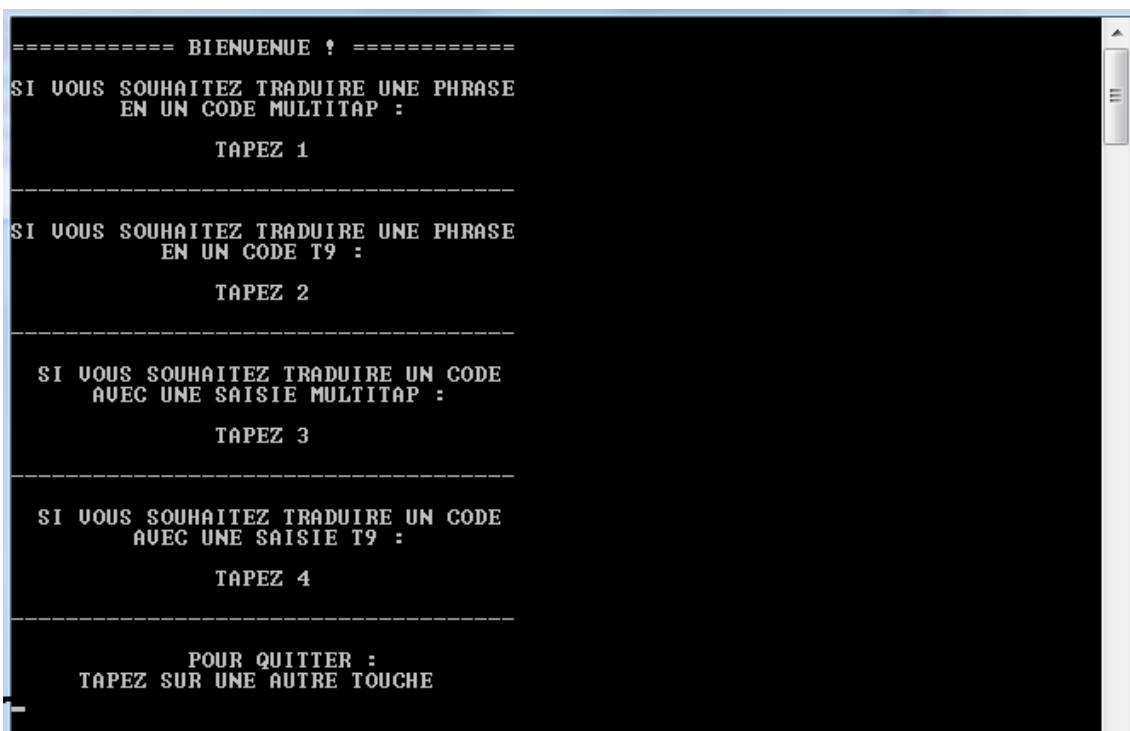
# 1. Notice utilisateur

## 1.1 Lancer le programme

Tout d'abord, vous devez lancer VisualStudio 2010, et ouvrir le programme « Projet\_S1.sln ». Appuyez simultanément sur les touches « Ctrl » et « F5 ».

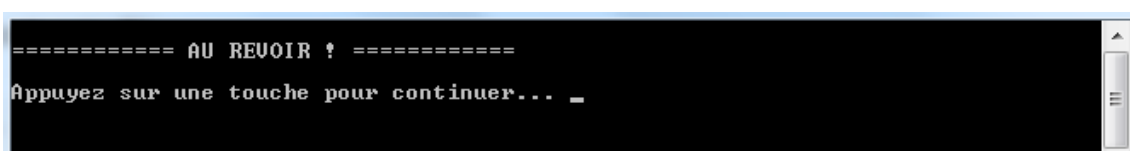
## 1.2 Naviguer entre les différentes fonctions : la page d'accueil

Vous arrivez sur la page d'accueil suivante :



```
===== BIENVENUE ? =====  
SI VOUS SOUHAITEZ TRADUIRE UNE PHRASE  
  EN UN CODE MULTITAP :  
      TAPEZ 1  
-----  
SI VOUS SOUHAITEZ TRADUIRE UNE PHRASE  
  EN UN CODE T9 :  
      TAPEZ 2  
-----  
SI VOUS SOUHAITEZ TRADUIRE UN CODE  
  AVEC UNE SAISIE MULTITAP :  
      TAPEZ 3  
-----  
SI VOUS SOUHAITEZ TRADUIRE UN CODE  
  AVEC UNE SAISIE T9 :  
      TAPEZ 4  
-----  
      POUR QUITTER :  
      TAPEZ SUR UNE AUTRE TOUCHE
```

Vous devez donc appuyer sur la touche 1, 2, 3 ou 4 pour accéder à l'une des quatre fonctions. La frappe de n'importe quelle autre touche du clavier, entraîne un message de fermeture et la fin du programme.



```
===== AU REVOIR ? =====  
Appuyez sur une touche pour continuer... _
```

### 1.3 Utiliser la première fonction : tapez 1

Si vous choisissez la première fonction, le message suivant s'affiche sur la console :

```
===== CODAGE MULTITAP =====  
  
/!\ VEUILLEZ LIRE ATTENTIVEMENT /!\  
LE MESSAGE CI-DESSOUS  
  
UNE FOIS LA LECTURE TERMINÉE,  
APPUYEZ SUR UNE TOUCHE DU CLAVIER  
  
===== NOTICE =====  
  
SAISISSEZ VOTRE PHRASE EN MAJUSCULES  
  
IL EST ÉGALEMENT POSSIBLE D'ENTRER  
DES CARACTÈRES ACCENTUÉS, DES CHIFFRES  
ET DES SIGNES DE PONCTUATION  
  
UNE FOIS LA SAISIE MULTITAP TERMINÉE  
APPUYEZ SUR LA TOUCHE "ENTRÉE"
```

Cela permet de décrire, globalement, ce que vous devez faire pour un fonctionnement optimal de la fonction.

Ensuite, vous devez appuyer sur n'importe quelle touche du clavier pour passer ce message et arriver sur l'interface de la fonction. La console affiche alors le message suivant :

```
Entrer la phrase à encoder en MultiTap :
```

Vous devez alors saisir, en majuscules, la phrase que vous aimeriez voir coder en une combinaison MultiTap.



Les signes de ponctuation « point », « virgule », « deux points », « point virgule », « point d'exclamation », « point d'interrogation », « apostrophe », « parenthèse ouvrante », « parenthèse fermante » et « trait d'union » sont autorisés.

Vous pouvez également entrer des chiffres, des espaces et des caractères majuscules accentués.



Il est possible, lors de la saisie, d'effacer le texte puis de réécrire autre chose.

Une fois la saisie de la phrase terminée, vous devez taper sur la touche « Entrée » de votre clavier pour valider votre saisie.

Le code s'affiche alors en-dessous de la phrase que vous venez de taper.

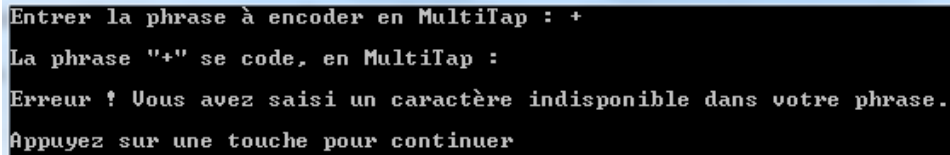
Exemple d'application :

```
Entrer la phrase à encoder en MultiTap : QUELLE HEURE EST-IL ?  
La phrase "QUELLE HEURE EST-IL ?" se code, en MultiTap :  
778833555*555330443388777330337777811144455501111  
Appuyez sur une touche pour continuer
```

En bas de la console, s'affiche un message de sortie (cf. image ci-dessus). Il faut donc appuyer sur n'importe quelle touche du clavier pour sortir de la fonction. Vous retombez alors sur la page

d'accueil.

Si lors de votre saisie, vous entrez un caractère qui n'est pas pris en charge par la fonction, le message suivant s'affichera lors de la validation :

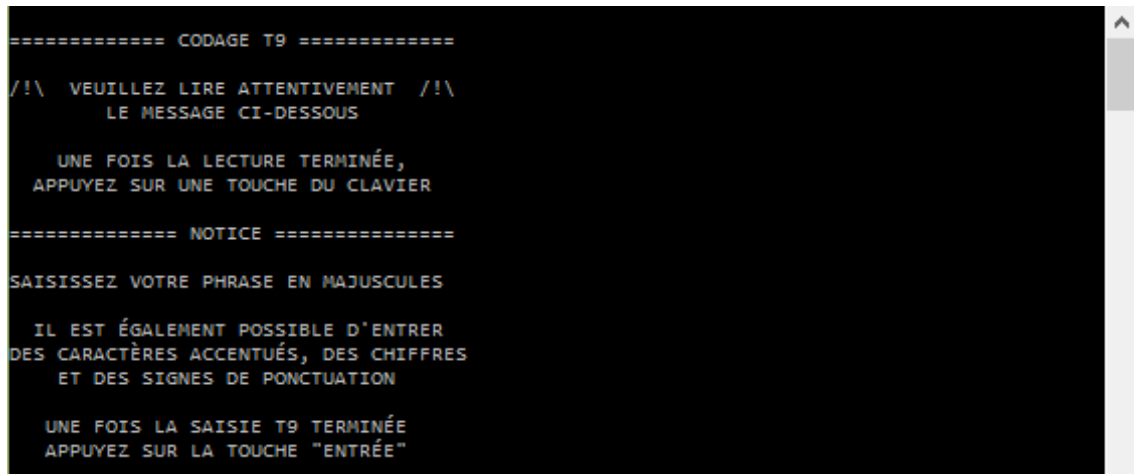


```
Entrer la phrase à encoder en MultiTap : +  
La phrase "+" se code, en MultiTap :  
Erreur ! Vous avez saisi un caractère indisponible dans votre phrase.  
Appuyez sur une touche pour continuer
```

Vous devez alors appuyer sur n'importe quelle touche du clavier et vous retournez sur la page d'accueil.

### 1.4 Utiliser la deuxième fonction : tapez 2

Si vous choisissez la deuxième fonction, le message suivant s'affiche sur la console :



```
===== CODAGE T9 =====  
/!\ VEUILLEZ LIRE ATTENTIVEMENT /!\  
LE MESSAGE CI-DESSOUS  
  
UNE FOIS LA LECTURE TERMINÉE,  
APPUYEZ SUR UNE TOUCHE DU CLAVIER  
  
===== NOTICE =====  
SAISISSEZ VOTRE PHRASE EN MAJUSCULES  
  
IL EST ÉGALEMENT POSSIBLE D'ENTRER  
DES CARACTÈRES ACCENTUÉS, DES CHIFFRES  
ET DES SIGNES DE PONCTUATION  
  
UNE FOIS LA SAISIE T9 TERMINÉE  
APPUYEZ SUR LA TOUCHE "ENTRÉE"
```

Celui-ci permet de donner, rapidement, les instructions que vous devez suivre pour le bon déroulement de la fonction.

Ensuite, vous devez appuyer sur n'importe quelle touche du clavier pour passer ce message et arriver sur l'interface de la fonction. La console affiche alors le message suivant :



```
Entrer la phrase à encoder en T9 :
```

Vous devez alors saisir, en majuscules, la phrase que vous souhaiteriez voir codée en une combinaison T9.



Les signes de ponctuation « point », « virgule », « deux points », « point virgule », « point d'exclamation », « point d'interrogation », « apostrophe », « parenthèse ouvrante », « parenthèse fermante » et « trait d'union » sont autorisés.

Vous pouvez également entrer des chiffres, des espaces et des caractères majuscules accentués.



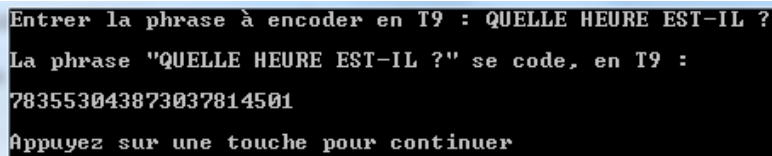
Il est possible, lors de la saisie, d'effacer le texte puis de réécrire autre chose.

## Le codage T9

Une fois la saisie de la phrase terminée, vous devez taper sur la touche « Entrée » de votre clavier pour valider votre saisie.

Le code s'affiche alors en-dessous de la phrase que vous venez de taper.

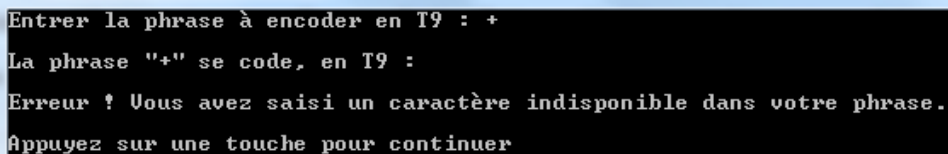
Exemple d'application :



```
Entrer la phrase à encoder en T9 : QUELLE HEURE EST-IL ?
La phrase "QUELLE HEURE EST-IL ?" se code, en T9 :
783553043873037814501
Appuyez sur une touche pour continuer
```

En bas de la console, s'affiche un message de sortie (cf. image ci-dessus). Il faut donc appuyer sur n'importe quelle touche du clavier pour sortir de la fonction. On retombe alors sur la page d'accueil.

Si lors de sa saisie, vous entrez en caractère qui n'est pas pris en charge par la fonction, le message suivant s'affichera lors de la validation :




```
Entrer la phrase à encoder en T9 : +
La phrase "+" se code, en T9 :
Erreur ! Vous avez saisi un caractère indisponible dans votre phrase.
Appuyez sur une touche pour continuer
```

Vous devez alors appuyer sur n'importe quelle touche du clavier et vous retournez sur la page d'accueil.

### 1.5 Utiliser la troisième fonction : tapez 3

Si vous choisissez la troisième fonction, le message suivant s'affiche sur la console :



```
===== DÉCODAGE MULTITAP =====
/!\  VEUILLEZ LIRE ATTENTIVEMENT  /!\
    LE MESSAGE CI-DESSOUS

    UNE FOIS LA LECTURE TERMINÉE,
    APPUYEZ SUR UNE TOUCHE DU CLAVIER

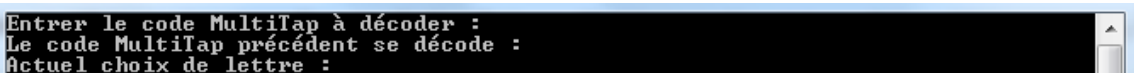
===== NOTICE =====
SAISISSEZ VOTRE CODE AVEC LES TOUCHES
    DU CLAVIER NUMÉRIQUE

    POUR MODIFIER VOTRE CHOIX :
    APPUYEZ SUR LA FLÈCHE DU BAS

UNE FOIS LA SAISIE MULTITAP TERMINÉE
    APPUYEZ SUR LA TOUCHE "ENTRÉE"
```

Cela permet de décrire, globalement, ce que vous devez faire pour un fonctionnement optimal de la fonction.

Ensuite, vous devez appuyer sur n'importe quelle touche du clavier pour passer ce message et arriver sur l'interface de la fonction. La console affiche alors le message suivant :

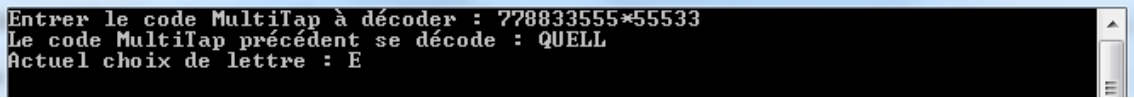


```
Entrer le code MultiTap à décoder :
Le code MultiTap précédent se decode :
Actuel choix de lettre :
```

## Le codage T9

La première ligne est destinée à recevoir le code que vous entrez, au fur et à mesure. La deuxième affiche, caractère par caractère, la traduction du code MultiTap que vous entrez. La troisième vous permet de visualiser votre actuel choix de caractère.

Exemple d'application :




```
Entrer le code MultiTap à décoder : 778833555*55533
Le code MultiTap précédent se decode : QUELL
Actuel choix de lettre : E
```



Vous pouvez appuyer sur toutes les touches numérotées, ainsi que sur l'astérisque. Les touches 2 à 9 décrivent les 26 lettres de l'alphabet, et incluent également les caractères accentués et les chiffres correspondant au numéro de la touche. La touche 1 permet d'avoir les signes de ponctuation et le chiffre 1. La touche 0 permet d'afficher un espace ou le chiffre 0. Il suffit d'appuyer autant de fois que nécessaire sur la touche portant le caractère souhaité pour l'obtenir. L'astérisque ne doit être employée qu'entre la saisie successive de deux lettres appartenant à la même touche.

Si vous n'appuyez pas sur un chiffre ou sur l'astérisque, le message d'erreur suivant s'affichera sur la console :



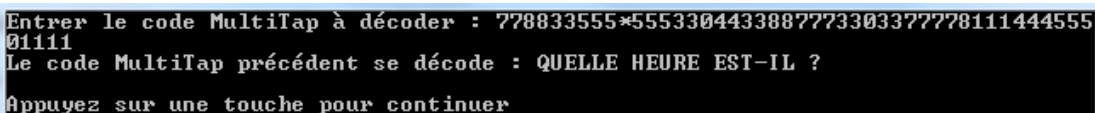
```
=====ERREUR DE SAISIE=====
      IL FAUT SAISIR DES CHIFFRES
      OU APPUYER SUR ENTRÉE POUR QUITTER
-----
Appuyez sur une touche pour continuer
```



À noter que, appuyer sur une autre touche que celle sur laquelle on appuyait précédemment, revient toujours à valider la lettre affichée sur la troisième ligne de la console. Ceci est valable sur les touches « autorisées », mais aussi sur les autres touches. Par ailleurs, une fois une lettre validée, il n'est pas possible de revenir en arrière. Il ne faut donc pas se tromper de touche lorsque vous voulez saisir un caractère !

Une fois que vous avez terminé de saisir votre code, vous devez appuyer sur « Entrée ». La console n'affiche alors plus que le code en MultiTap que vous venez de taper, et sa traduction.

Exemple d'application :



```
Entrer le code MultiTap à décoder : 778833555*5553304433887773303377778111444555
01111
Le code MultiTap précédent se decode : QUELLE HEURE EST-IL ?
Appuyez sur une touche pour continuer
```

En bas de la console, s'affiche un message de sortie (cf. image ci-dessus). Il faut donc appuyer sur n'importe quelle touche du clavier pour sortir de la fonction. On retombe alors sur la page d'accueil.



### 1.6 Utiliser la quatrième fonction : tapez 4

Si vous choisissez la quatrième fonction, le message suivant s'affiche sur la console :

```
===== DÉCODAGE T9 =====
/!\  UEUILLEZ LIRE ATTENTIVEMENT  /!\
      LE MESSAGE CI-DESSOUS

      UNE FOIS LA LECTURE TERMINÉE,
      APPUYEZ SUR UNE TOUCHE DU CLAVIER

===== NOTICE =====
SAISISSEZ VOTRE CODE AVEC LES TOUCHES
      DU CLAVIER NUMÉRIQUE

      POUR MODIFIER VOTRE CHOIX :
      APPUYEZ SUR LA FLECHE DU BAS

LORSQU'UN MOT N'EST PAS DANS LE DICO
      APPUYEZ SUR LA TOUCHE "INSÉRER"

UNE FOIS LA SAISIE MULTITAP TERMINÉE
      APPUYEZ SUR LA TOUCHE "ENTRÉE"

QUAND VOUS AVEZ TERMINÉ LE DÉCODAGE,
      APPUYEZ SUR LA TOUCHE "ENTRÉE"
```

Celui-ci permet de donner, rapidement, les instructions que vous devez suivre pour le bon déroulement de la fonction.

Ensuite, vous devez appuyer sur n'importe quelle touche du clavier, et la console affiche le message suivant :

```
Votre code :
Votre phrase :
Votre choix :
```

La première ligne est destinée à recevoir le code que vous entrez, au fur et à mesure. La deuxième affiche, mot par mot, la traduction du code T9 que vous entrez. La troisième vous permet de visualiser votre actuel choix de mot.

Exemple d'application :

```
Votre code : 78355304
Votre phrase : QUELLE
Votre choix : G_
```



Vous ne pouvez appuyer que sur les touches numérotées.

Si vous n'appuyez pas sur un chiffre, le message d'erreur suivant s'affichera sur la console :

```
=====ERREUR DE SAISIE=====

      IL FAUT SAISIR DES CHIFFRES
      OU APPUYER SUR ENTRÉE POUR QUITTER

-----
Appuyez sur une touche pour continuer
```

## Le codage T9

Il suffit alors d'appuyer sur n'importe quelle touche du clavier pour revenir à l'interface de la fonction.



- Les touches 2 à 9 décrivent les 26 lettres de l'alphabet.
- La touche 1 permet d'avoir les signes de ponctuation.
- La touche 0 permet d'afficher un espace.

Au fur et à mesure que vous entrez votre code en T9, la fonction vous propose automatiquement de choisir un mot, sur la dernière ligne. Vous ne devez pas vous en soucier, sauf si vous avez terminé de taper le mot que vous souhaitiez écrire. Dans ce cas, vous pouvez vous retrouver face à deux situations différentes.

Si vous avez tapé un code qui correspond à un ou plusieurs mots dans le dictionnaire, il suffit que vous appuyiez sur la flèche du bas pour faire défiler les possibilités. Ces possibilités défilent sur la troisième ligne. (Dans le cas, où le code ne correspond qu'à un seul mot, le même mot défile sur la console.) Lorsque vous voyez apparaître le mot qui vous convient, vous n'avez plus qu'à continuer de taper son code.

Exemple :

```
Votre code : 783553043873
Votre phrase : QUELLE
Votre choix : HETRE

Votre code : 783553043873
Votre phrase : QUELLE
Votre choix : HEURE
```



Attention ! N'appuyez pas sur « Entrée », sinon cela sera considéré comme la fin de votre phrase, et vous quitteriez alors la fonction.

Si vous avez tapé un code qui ne correspond à aucun mot dans le dictionnaire, la console affiche le message suivant :

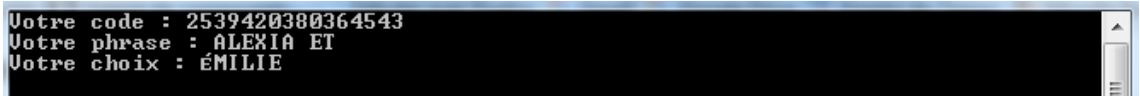
```
Votre code : 78355
Votre phrase :
Votre choix : QUELJ
Ce mot n'existe pas dans le dictionnaire.
Appuyez sur la touche Insérer si vous avez fini d'écrire votre mot et qu'il n'appartient au dictionnaire.
Sinon, continuer à entrer votre code.
```

Vous devez donc appuyer sur la touche « Inser » de votre clavier, puis entrer en MultiTap le mot que vous vouliez écrire. Une fois le mot saisi, il suffit que vous appuyiez sur la touche « Entrée », puis que vous continuez de taper votre code.

Exemple :

```
Votre code : 2539420380364543
Votre phrase : ALEXIA ET
Votre choix : FOIJGD
Ce mot n'existe pas dans le dictionnaire.
Appuyez sur la touche Insérer si vous avez fini d'écrire votre mot et qu'il n'appartient au dictionnaire.
Sinon, continuer à entrer votre code.
```

## Le codage T9



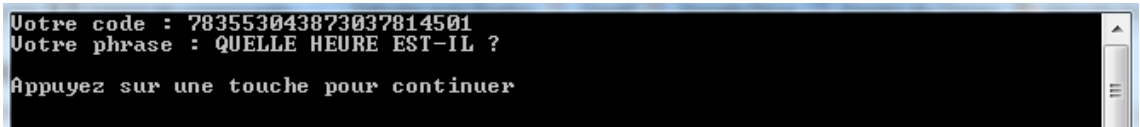
```
Votre code : 2539420380364543
Votre phrase : ALEXIA ET
Votre choix : EMILIE
```



Attention ! Il n'est ni possible d'effacer, dans la première ligne de la console, un caractère du code T9 déjà saisi, ni de modifier, dans la deuxième ligne, un mot de la phrase en construction déjà validé.

Une fois que vous avez terminé de saisir votre code en T9, vous devez appuyer sur « Entrée ». La console n'affiche alors plus que votre code entré en T9, et sa traduction.

Exemple d'application :



```
Votre code : 783553043873037814501
Votre phrase : QUELLE HEURE EST-IL ?
Appuyez sur une touche pour continuer
```

En bas de la console, s'affiche un message de sortie (cf. image ci-dessus). Il faut donc appuyer sur n'importe quelle touche du clavier pour sortir de la fonction. On retombe alors sur la page d'accueil.

### 1.7 Remarques

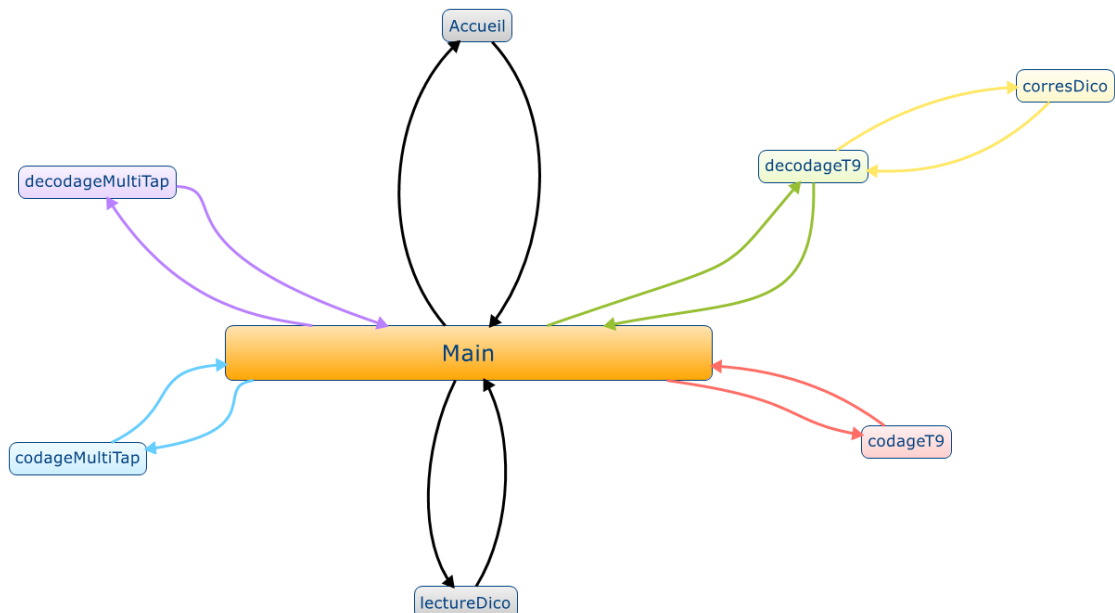
Vous pouvez utiliser la première et la deuxième fonction, pour tester, respectivement, la troisième et la quatrième fonction. Pour cela, il suffit de vous munir d'un crayon et d'un papier, et de recopier les codes générés par la première (ou la deuxième) fonction, afin de les tester dans la troisième (ou la quatrième) fonction.

Pour entrer les codes de la troisième et de la quatrième fonction, il est possible d'utiliser les touches chiffrées du pavé numérique, mais aussi les touches numérotées au-dessus des lettres. Pour utiliser ces dernières, il n'est pas nécessaire d'appuyer sur la touche « Shift » au préalable.

## 2. Structuration des données

### 2.1 Architecture

Notre programme comporte 6 fonctions et une procédure, dont voici l'agencement :



Lors du lancement du programme, le « Main » fait appel successivement à deux fonctions. Dans un premier temps, il va faire appel à la fonction « lectureDico » pour ranger dans un tableau tous les mots du dictionnaire. Dans un second temps, il lance la fonction « Accueil ». Cette dernière renvoie au « Main » le clé correspondant au choix du chiffre de l'utilisateur.

Si l'utilisateur a appuyé sur la touche 1, 2, 3 ou 4, le « Main » appelle, respectivement, les fonctions « codageMultiTap », « codageT9 », « decodageMultiTap » ou « decodageT9 ». Une fois ces dernières exécutées, on retourne au « Main » qui lancera de nouveau la fonction « Accueil ». Si l'utilisateur appuie sur une touche différente de 1, 2, 3 et 4, on quitte le programme.

La fonction « decodageT9 » fait appel à la fonction « CorresDico » afin de faire correspondre le code entré par l'utilisateur aux mots du dictionnaire, préalablement rangé dans un tableau.



### Remarques

- On considère la touche 0 du téléphone comme l'espace.
- On considère la touche 1 du téléphone comme la ponctuation.
- On considère la touche \* du téléphone comme le séparateur de deux lettres appartenant à la même touche.
- Avant le démarrage de chaque fonction, les consignes seront données à l'utilisateur pour lui indiquer les règles à suivre pour un bon fonctionnement de la fonction.



Attention ! Dans notre rapport, on appelle :

- première fonction, la fonction `codageMultiTap`
- deuxième fonction, la fonction `codageT9`
- troisième fonction, la fonction `decodageMultiTap`
- quatrième fonction, la procédure `decodageT9`

## 2.2 Fonction « `lectureDico` »

### 2.2.1 Description

« `lectureDico` » est une fonction qui prend en argument une chaîne de caractères, et qui renvoie un tableau de chaînes de caractères.

L'argument correspond au nom du fichier source (ici le dictionnaire « `dicoFR.txt` » fourni par les enseignants) et le tableau renvoyé contient, dans chacune de ses cases, une des lignes du fichier source (ici, un unique mot).

### 2.2.2 Fonctionnement

On crée deux instances de `StreamReaders` qui permettront de parcourir deux fois le fichier source. On parcourt une première fois le fichier source pour compter le nombre de lignes (soit le nombre de mots). On crée alors un tableau comportant le même nombre de cases que le nombre de lignes du fichier source. Ensuite, on re-parcourt le fichier source, afin de ranger chacune de ses lignes (soit chacun des mots) dans les cases du tableau.

## 2.3 Fonction « `Accueil` »

### 2.3.1 Description

« `Accueil` » est une fonction qui ne prend rien en entrée et qui renvoie un `ConsoleKeyInfo`.

Elle permet d'afficher le message d'accueil de la fonction, et renvoie la clé de la touche correspondant au choix de l'utilisateur.

### 2.3.2 Fonctionnement

Un « `Console.WriteLine` » permet l'affichage du message d'accueil. On crée une variable de type « `ConsoleKeyInfo` » pour retourner le choix de l'utilisateur.

### 2.4 Fonction « `codageMultiTap` »

#### 2.4.1 Description

« `codageMultiTap` » est une fonction qui prend en argument une chaîne de caractères, et qui renvoie une chaîne de caractères.

L'argument correspond à la phrase que l'utilisateur veut coder en MultiTap, et la chaîne de caractères renvoyée est donc son code.

#### 2.4.2 Fonctionnement

On crée un tableau de chaînes de caractères « alphabet » de taille [10, 2], qui a chaque touche du clavier du téléphone associe les caractères possibles.

Ensuite, on parcourt, caractère par caractère, la chaîne entrée en argument grâce à une boucle « while ».

Pour chaque caractère de la chaîne entrée, on vérifie qu'il appartient à une touche du clavier du téléphone, en procédant comme il suit.

- On se « place » sur la première touche du téléphone, et on regarde si le caractère de la chaîne entrée correspond à un des caractères de cette touche.
- Si aucun caractère ne correspond, on se « place » sur la seconde touche, et on regarde si le caractère de la chaîne entrée, correspond à un des caractères de cette touche.
- Et ainsi de suite, jusqu'à il y ait une correspondance.

S'il n'y a pas de correspondances, on retourne un message d'erreur.

S'il y a une correspondance, on procède en deux fois.

- On vérifie, dans un premier temps, que la lettre précédente n'est pas sur la même touche que celle qu'on est en train de coder. Si c'est le cas, on ajoute à la chaîne à retourner le caractère « \* » pour bien préciser à l'utilisateur qu'il s'agit de deux lettres différentes.
- Dans un second temps, on ajoute à la chaîne à retourner la combinaison de chiffres associée à la lettre qu'on vient de coder.

Lorsqu'on a parcouru toute la chaîne de caractères entrée en argument, on retourne la chaîne que l'on a concaténée au fur et à mesure, soit le code en MultiTap correspondant.

### 2.5 Fonction « `codageT9` »

#### 2.5.1 Description

« `codageT9` » est une fonction qui prend en argument une chaîne de caractères, et qui renvoie une chaîne de caractères.

L'argument correspond à la phrase que l'utilisateur veut coder en T9, et la chaîne de caractères renvoyée est donc son code.

#### 2.5.2 Fonctionnement

On crée un tableau de chaînes de caractères « alphabet » de taille [10, 2], qui a chaque touche du clavier du téléphone associe les caractères possibles.

Ensuite, on parcourt, caractère par caractère, la chaîne entrée en argument grâce à une boucle « while ».

Pour chaque caractère de la chaîne entrée, on vérifie qu'il appartient à une touche du clavier du téléphone, en procédant comme il suit.

- On se « place » sur la première touche du téléphone, et on regarde si le caractère de la chaîne entrée correspond à un des caractères de cette touche.
- Si aucun caractère ne correspond, on se « place » sur la seconde touche, et on regarde si le caractère de la chaîne entrée correspond à un des caractères de cette touche.
- Et ainsi de suite, jusqu'à il y ait une correspondance.

S'il n'y a pas de correspondances, on retourne un message d'erreur.

S'il y a une correspondance, on ajoute à la chaîne à retourner le chiffre associé à la lettre qu'on vient de coder.

Lorsqu'on a parcouru toute la chaîne de caractères entrée en argument, on retourne la chaîne que l'on a concaténée au fur et à mesure, soit la code correspondant.

## 2.6 Fonction « `decodageMultiTap` »

### 2.6.1 Description

« `decodageMultiTap` » est une fonction qui ne prend rien en argument, et qui renvoie une chaîne de caractères.

Cette fonction permet d'afficher le code et la phrase, lettre par lettre, que l'utilisateur rentre au fur et à mesure en MultiTap.

La fonction finit par renvoyer la chaîne de caractères correspondant à la phrase entièrement décodée.

### 2.6.2 Fonctionnement

On crée un tableau de chaînes de caractères « alphabet » de taille [10, 2], qui a chaque touche du clavier du téléphone associe les caractères possibles.

Ensuite, on entre dans une boucle « do while » qui permet de décoder les chiffres entrés par l'utilisateur en leur associant les caractères correspondants, tant qu'il ne finit pas sa phrase (en appuyant sur « Entrée »).

La seconde boucle « while » permet de prendre en compte la répétition de la touche, propre au codage MultiTap. Lors de la saisie d'un caractère par l'utilisateur, quatre situations sont possibles.

- Si l'utilisateur appuie sur la même touche du clavier numérique que précédemment.  
Dans ce cas, la boucle « while » continue et compte le nombre de fois où l'utilisateur a appuyé sur cette touche. Cela permet de déterminer le caractère correspondant au choix de l'utilisateur et donc à l'afficher dans la console.  
Le « dépassement » est pris en compte, c'est-à-dire si l'utilisateur dépasse la lettre qui l'intéresse, il peut y réavoir accès en continuant d'appuyer sur le chiffre sur lequel se situe sa lettre.
- Si l'utilisateur appuie sur une nouvelle touche du clavier numérique.  
Dans ce cas, on quitte la boucle « while », et on ajoute le caractère précédent à la chaîne de caractères en construction.

- Si l'utilisateur appuie sur « \* » pour indiquer qu'il veut enchaîner deux lettres identiques. Dans ce cas, on quitte la boucle « while », et on ajoute le caractère précédent à la chaîne de caractères en construction.
- Si l'utilisateur appuie sur une touche du clavier non incluse dans le tableau « alphabet ». Dans ce cas, cela lui renvoie un message d'erreur.

À chaque fois que l'utilisateur appuie sur une touche « autorisée », sa saisie est ajoutée à une chaîne de caractères qui permet d'afficher le code entré par celui-ci.

La fonction finit par renvoyer la phrase correspondante, c'est-à-dire la chaîne de caractères construite petit à petit.

## 2.7 Fonction « **corresDico** »

### 2.7.1 Description

« **corresDico** » est une fonction qui prend en argument un tableau de chaînes de caractères et une chaîne de caractères, et renvoie un tableau de chaîne de caractères.

Les arguments correspondent, respectivement, à un tableau qui contient dans chacune de ses cases un mot du dictionnaire (tableau obtenu grâce à la fonction **lectureDico**) ; et au code auquel on veut faire correspondre des mots du tableau.

Le tableau renvoyé contient donc, dans chacune de ses cases, les mots correspondant au code pris en argument.

### 2.7.2 Fonctionnement

Tout d'abord, on crée un tableau de chaînes de caractères « **alphabet** » de taille [8, 2], qui a chaque touche du clavier du téléphone associe les lettres et chiffres possibles.

Cette fonction peut être séparée en trois étapes.

#### 1ère étape :

On fait correspondre chaque chiffre du code entré par l'utilisateur au groupe de lettres situé sur la touche portant ce chiffre. Cette étape se fait suite à la création d'un tableau de chaînes de caractères de la taille du code.

#### 2ème étape :

La première boucle « **for** » permet de parcourir l'ensemble du dictionnaire. Une première condition vient limiter le parcours : on vérifie si le mot du dictionnaire est de la même taille que la taille du code. Cela permet de restreindre les mots possibles et d'éviter des parcours inutiles.

Ensuite, une fois que la taille du mot correspond, une boucle « **while** » permet de vérifier si chaque lettre du mot du dictionnaire correspond à une des lettres du paquet de lettres du tableau créé précédemment, sachant que la position du paquet de lettres du tableau correspond à la position de la lettre du dictionnaire.

Si un mot du tableau en argument correspond au code en argument, c'est-à-dire s'il a la même longueur et que chacune de ses lettres appartient aux paquets de lettres du tableau construit à la 1ère étape, on le concatène à la chaîne de caractères des mots compatibles, en ajoutant un espace, afin que le prochain mot compatible ne soit pas « collé » au mot qu'on vient d'ajouter. De plus, on incrémente de un, un compteur permettant de connaître le nombre de mots compatibles.



### 3ème étape :

On crée un tableau des mots compatibles grâce à la chaîne de caractères construite à la 2ème étape. Pour cela, on crée tout d'abord un tableau de la taille du nombre de mots compatibles, déterminé également lors de la 2ème étape.

Ensuite, on parcourt la chaîne de caractères des mots possibles, et tant qu'on ne rencontre pas un espace, on ajoute, une à une, les lettres de la chaîne de caractères des mots compatibles dans une case du tableau. Lorsqu'on rencontre un espace, cela signifie qu'on vient de placer dans cette case toutes les lettres d'un mot. Il faut donc changer de case, et reprendre le parcours de la chaîne. Le processus se termine lorsque toutes les cases du tableau sont remplies.

La fonction renvoie donc les mots dans un tableau qui correspondent au code entré en paramètre s'il y en a. Dans le cas contraire, la fonction renvoie un tableau d'une seule case, cette dernière contenant un simple espace.

## 2.8 Procédure « decodageT9 »

### 2.8.1 Description

« decodageT9 » est une procédure qui prend en argument un tableau de chaînes de caractères.

L'argument est le dictionnaire préalablement trié dans un tableau, et la procédure affiche au fur et à mesure, la phrase correspondant au code-chiffres rentré par l'utilisateur.

### 2.8.2 Fonctionnement

On entre dans une boucle « do while » qui permet d'associer les chiffres entrés par l'utilisateur à des mots, tant qu'il ne finit pas sa phrase (en appuyant sur « Entrée »).

La seconde boucle « while » permet de déterminer le mot souhaité par l'utilisateur en récupérant les chiffres (2, 3, 4, 5, 6, 7, 8, 9) entrés par celui-ci. Tant qu'il appuie sur les touches correspondant à ces chiffres, cela signifie qu'il continue d'écrire un mot.

Lors de la saisie d'un chiffre par l'utilisateur, le code en construction entré par l'utilisateur correspond à aucun, un ou plusieurs mots du dictionnaire (cette étape se fait grâce à l'appel de la fonction « corresDico »). Dans ce cas là, les mots possibles se trouvent dans un tableau de chaînes de caractères. Deux situations sont possibles.

- Si le code ne correspond à aucun mot du dictionnaire.  
Dans ce cas, la console affiche le premier caractère correspondant à la touche du clavier, et le mot en construction est également incrémenté de ce caractère.  
Si l'utilisateur n'a pas fini de rentrer son code, l'utilisateur peut continuer d'entrer son code, et la fonction continue normalement.  
En revanche, s'il a fini de rentrer son code, et qu'aucun mot n'existe, si celui-ci appuie sur « Inser », il peut rentrer son mot grâce à la fonction `decodageMultiTap`. Cet appel à la fonction permet de récupérer le mot (une chaîne de caractères) souhaité.
- Si le code rentré par l'utilisateur correspond à un ou plusieurs mots du dictionnaire.  
Dans ce cas, la console affiche le premier mot correspondant au code en construction, et le mot en construction prend également ce mot comme valeur.  
Si l'utilisateur n'a pas fini de rentrer son code, l'utilisateur peut continuer d'entrer son code, et la fonction continue.  
En revanche si celui-ci a fini, alors, si le premier mot du tableau de mots possibles ne

lui convient pas, il peut faire défiler les autres mots en appuyant sur la flèche du bas. Le « dépassement » est pris en compte, c'est-à-dire si l'utilisateur dépasse le mot qui l'intéresse, il peut y réavoir accès en continuant d'appuyer sur la flèche du bas.

Si l'utilisateur entre un code différent de 2, 3, 4, 5, 6, 7, 8, 9, il y a trois autres possibilités.

- Si celui-ci appuie sur 0. Cela signifie donc, qu'il a fini d'écrire un mot, et qu'il désire faire un espace. On incrémente donc la chaîne de caractères correspondant à la phrase, avec le mot dont on vient de taper le code, et on réinitialise la chaîne de caractères correspondant au code et au mot en construction.
- Si celui-ci appuie sur 1. Cela signifie donc qu'il a fini d'écrire un mot, et qu'il désire écrire un signe de ponctuation (la variable correspondante est la chaîne de caractères « ponctuation »). On incrémente donc la chaîne de caractères correspondant à la phrase, avec le mot dont on vient de taper le code, et on réinitialise la chaîne de caractères correspondant au code et au mot en construction. Ensuite, par défaut, on propose à l'utilisateur un point, mais tant qu'il appuie sur la flèche du bas, il pourra choisir un autre signe de ponctuation.
- Si celui-ci appuie sur une touche différente du clavier numérique. Dans ce cas là, la console affiche un message d'erreur, et l'utilisateur a simplement à appuyer sur une touche, pour retourner sur l'interface de la procédure.

À chaque fois que l'utilisateur appuie sur une touche « autorisée », sa saisie est ajoutée à une chaîne de caractères qui permet d'afficher le code global entré par celui-ci.

La procédure finit par afficher la phrase correspondante au code entré par l'utilisateur.

## 3. Exemples

Voici les résultats que l'on obtient en faisant fonctionner la fonction « Accueil », la première, la deuxième, la troisième et la quatrième fonction. Ces exemples sont repris de la notice utilisateur.

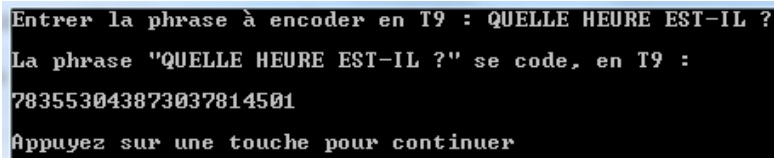
### 3.1 Fonction « Accueil »

```
===== BIENVENUE ! =====  
SI VOUS SOUHAITEZ TRADUIRE UNE PHRASE  
  EN UN CODE MULTITAP :  
    TAPEZ 1  
-----  
SI VOUS SOUHAITEZ TRADUIRE UNE PHRASE  
  EN UN CODE T9 :  
    TAPEZ 2  
-----  
SI VOUS SOUHAITEZ TRADUIRE UN CODE  
  AVEC UNE SAISIE MULTITAP :  
    TAPEZ 3  
-----  
SI VOUS SOUHAITEZ TRADUIRE UN CODE  
  AVEC UNE SAISIE T9 :  
    TAPEZ 4  
-----  
POUR QUITTER :  
  TAPEZ SUR UNE AUTRE TOUCHE
```

### 3.2 Fonction « codageMultiTap »

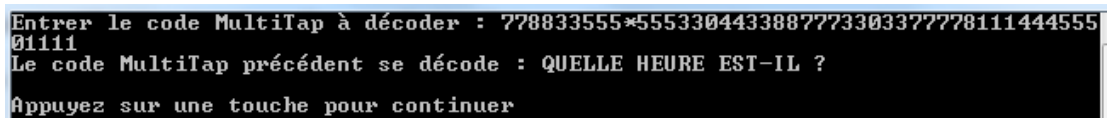
```
Entrer la phrase à encoder en MultiTap : QUELLE HEURE EST-IL ?  
La phrase "QUELLE HEURE EST-IL ?" se code, en MultiTap :  
778833555*555330443388777330337777811144455501111  
Appuyez sur une touche pour continuer
```

### 3.3 Fonction « codageT9 »



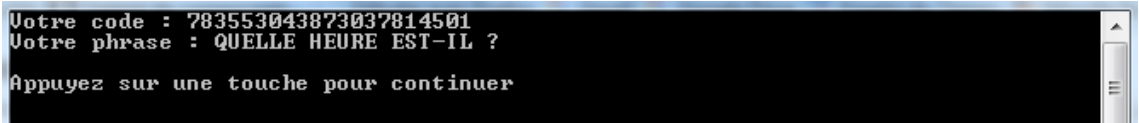
```
Entrer la phrase à encoder en T9 : QUELLE HEURE EST-IL ?  
La phrase "QUELLE HEURE EST-IL ?" se code, en T9 :  
783553043873037814501  
Appuyez sur une touche pour continuer
```

### 3.4 Fonction « decodageMultiTap »



```
Entrer le code MultiTap à décoder : 778833555*5553304433887773303377778111444555  
01111  
Le code MultiTap précédent se décode : QUELLE HEURE EST-IL ?  
Appuyez sur une touche pour continuer
```

### 3.5 Procédure « decodageT9 »



```
Votre code : 783553043873037814501  
Votre phrase : QUELLE HEURE EST-IL ?  
Appuyez sur une touche pour continuer
```



## 4. Idées et évolutions

### 4.1 Idées

Plusieurs tentatives ont été faites au cours du projet.

On pensait utiliser une énumération au lieu d'un tableau [,] pour l'alphabet. En revanche, faire comprendre à l'ordinateur que la position dans l'énumération d'une lettre correspondait à entier n'était pas possible. En effet, lorsqu'on forçait la lettre de l'énumération à devenir en entier, il renvoyait le code ASCII.

Avant de proposer une console dynamique, on proposait à l'utilisateur le choix des fonctions, et de rentrer un seul code/mot, et on lui renvoyait le résultat attendu. On a donc ensuite évolué en proposant à l'utilisateur la ponctuation et les caractères accentués ainsi que les chiffres. Dans la même lancée, on a proposé à l'utilisateur une interface dynamique pour la troisième et la quatrième fonction, où il rentre petit à petit les caractères, et les choix s'offrent petit à petit à lui. D'où le résultat maintenant.

Les principales évolutions se sont faites autour de la procédure « `decodageT9` ».

Au départ, on faisait les propositions à l'utilisateur des mots correspondant au code qu'il avait entré, sans dynamisme. Pour cela, on avait une fonction qui créait un fichier texte qui récupérait les mots du dictionnaire ayant une taille identique au code rentré par l'utilisateur, et on les copiait dans ce nouveau fichier. Et par la suite, on cherchait dans ce fichier texte les mots qui pouvaient correspondre au code. On créait alors un nouveau fichier texte qui comprenait la liste des mots compatibles au code.

Une fois cette fonction établie, on a tenté de proposer à l'utilisateur d'écrire des phrases. L'utilisateur entrait son code, et à chaque partie du code entré (à la limite du 0, correspondant à l'espace) correspondait un mot. On faisait donc appel à cette fonction à chaque mot.

À chaque appel de cette fonction, ouvrir le fichier texte du dictionnaire, et créer à chaque fois deux nouveaux fichiers texte pouvait prendre du temps si la phrase était longue, ce n'était donc pas optimal. On a donc décidé de ranger les mots du dictionnaire dans un tableau pour éviter d'ouvrir le fichier texte plusieurs fois, et de regrouper la fonction de vérification de la longueur avec la vérification du code, caractère par caractère.

À chaque mot « code » de la phrase « code » entré par l'utilisateur, on affichait, sous forme de liste numérotée les mots du dictionnaire compatibles à ce mot « code ». L'utilisateur devait alors taper le chiffre correspondant à son choix.

On a ensuite proposé une interface dynamique où l'on pouvait faire défiler les mots avec la flèche du bas, et où le dépassement est géré, c'est-à-dire, si l'utilisateur rate le mot, il peut continuer à appuyer sur la flèche du bas pour retrouver son mot.

### 4.2 Évolutions

Nous avons pensé à plusieurs évolutions pour le programme que nous avons écrit. En voici la liste.

#### 4.2.1 Pour la première et la deuxième fonction

##### **Coder davantage de caractères.**

Il n'est actuellement pas possible à l'utilisateur de rentrer des phrases comportant des lettres minuscules, ni certains caractères, comme les signes mathématiques (+, =, <, >, %, etc.), les guillemets, les devises (€, \$, £, etc.)... Il serait donc intéressant d'améliorer le programme, afin que ses caractères soient accessibles à l'utilisateur.

#### 4.2.2 Pour la troisième et la quatrième fonction

##### **Effacer la saisie.**

Les fonctions de décodage ont un inconvénient. Il n'est pas possible d'effacer le code que l'on vient d'écrire. Ainsi, si à un moment ou un autre, on se rend compte qu'on a tapé sur la mauvaise touche du clavier, il est impossible de faire machine arrière. L'évolution qui consisterait à permettre à l'utilisateur d'effacer sa saisie serait donc un atout considérable pour notre programme.

##### **Écrire davantage de caractères.**

Avec les seules touches numérotées (et l'astérisque), il n'est pas possible à l'utilisateur d'écrire des phrases comportant certains caractères, notamment les lettres minuscules, les symboles mathématiques, les guillemets et les devises. Ainsi, il serait intéressant de trouver une solution permettant à l'utilisateur d'écrire ses caractères là. Par exemple, en utilisant la touche dièse du clavier qui n'est pour l'instant pas exploitée, on pourrait basculer de majuscules à minuscules.

#### 4.2.3 Pour la fonction `decodageMultiTap`

##### **Gérer plus efficacement les erreurs.**

Si jamais l'utilisateur tape, par inadvertance, sur une touche non autorisée, cela valide automatiquement la lettre qu'il était en train d'écrire. Ainsi, s'il veut écrire un « R » et qu'il appuie deux fois sur la touche 7, avant de malencontreusement taper sur la touche « Effacer », cela lui valide un « Q ». Et lorsqu'il reviendra sur l'interface, il ne lui sera pas possible de réappuyer sur 7 pour former son R. Il serait donc appréciable de remédier à cette situation, en ne validant une lettre que si l'utilisateur tape sur une nouvelle touche autorisée.

#### 4.2.4 Pour la procédure `decodageT9`

##### **Épeler des mots en MultiTap.**

Si l'utilisateur est en mesure de rentrer un nouveau mot en MultiTap, en tapant sur la touche « Inser », cela n'est possible que si le code qu'il a entré ne correspond à aucun mot dans le dictionnaire. Ainsi, une évolution possible serait de permettre à l'utilisateur d'épeler en MultiTap un mot, quel que soit le nombre de mots du dictionnaire compatibles avec son code.

##### **Vérification de la cohérence de MultiTap et T9.**

Lorsque l'utilisateur entre un code qui ne correspond à aucun mot dans le dictionnaire, il peut épeler un mot en MultiTap, s'il le souhaite. Néanmoins, à aucun moment on ne vérifie que le mot

entré en MultiTap correspond au code incompatible avec les mots du dictionnaire, c'est-à-dire, que si l'utilisateur entre un code de 3 lettres qui n'existe pas, il pourra rentrer un mot de 6 lettres, et aucune vérification ne sera faite. Il serait donc judicieux de créer une fonction qui s'assure que l'utilisateur entre en MultiTap un code cohérent avec celui écrit en T9.

### **Ajouter des mots au dictionnaire.**

Lorsque l'utilisateur entre un code qui ne correspond à aucun mot dans le dictionnaire, on lui propose d'épeler son mot en MultiTap. Malheureusement, ce mot n'est pas sauvegardé dans le dictionnaire. Ainsi, à chaque fois que l'utilisateur voudra réécrire ce mot, il devra de nouveau devoir l'épeler. Il serait donc judicieux d'enregistrer automatiquement les mots ajoutés par l'utilisateur en MultiTap dans le dictionnaire.

### **Utiliser toutes les flèches directrices.**

Pour l'instant, lorsque l'utilisateur veut faire défiler les mots compatibles à son code (en construction), il ne peut qu'utiliser la flèche du bas. Il serait sûrement intéressant de prendre également en compte dans le programme, l'utilisation des autres flèches directrices, afin d'éviter des erreurs de frappe.