

**Auteur : Marco Lavoie****Instructeur:** Sébastien Bois

Date de distribution	Groupe	Échéancier
8 février 2021	IFM025908-010	22 février 2021 à 23h59
10 février 2021	IFM025908-020	24 février 2021 à 23h59

**Devoir No.4****Notes sur les fichiers soumis**

Les devoirs remis en retard ne seront pas corrigés.

Vous devez soumettre via eCité un fichier source  
nom\_prenom\_devoir\_4.cpp.

L'instructeur a présenté en classe des bonnes pratiques de programmation en C++ que des conventions d'écriture. Assurez-vous d'appliquer des pratiques et conventions pour ne pas voir votre travail pénalisé inutilement.

**À effectuer**

Parmi tous les arrangements de codage, le plus célèbre est certainement le **code Morse**, développé par *Samuel Morse* en 1832, utilisé avec le système de télégraphe. Le code Morse affecte une série de points (appelés *ti*) et de traits (appelés *ta*) à chaque lettre de l'alphabet, à chaque chiffre et à quelques caractères spéciaux (comme le point, la virgule, le deux-points et le point-virgule). Pour les systèmes qui impliquent un moyen sonore, le point représente un son court tandis que le trait représente un son long. D'autres représentations des points et des traits sont utilisés avec des systèmes utilisant des moyens lumineux ou à drapeaux.

La séparation entre deux mots est indiquée par un espace, un barre oblique (/) ou tout simplement par l'absence de point ou de trait. Dans un système utilisant un moyen sonore, un espace s'indique par une courte pause pendant laquelle aucun son n'est transmis.

La version internationale du code Morse est illustrée ci-dessous. Un caractère traduisible peut contenir un maximum 6 ti(points) et ta(trait) :

<u>Caractère</u>	<u>Code</u>	<u>Caractère</u>	<u>Code</u>	<u>Caractère</u>	<u>Code</u>
A	.-	N	--.	0	-----
B	....	O	---	1	.----
C	..--	P	..--.	2	..---
D	---.	Q	-----	3	...--
E	.	R	..-	4	....-
F	..--	S	...	5	.....
G	--.	T	-	6	-....
H	....	U	..-	7	--...
I	..	V	...-	8	---..
J	....	W	..-	9	----.

K	--	X	---	.	.-.-.
L	....	Y	----	,	--..--
M	--	Z	----	;	-.-.;

Écrivez un programme qui lit une phrase écrite en français (sans accents) et qui code celle-ci en langage Morse visuellement (sous forme de points et de traits) dans la console, ainsi que de façon sonore à l'aide du haut-parleur de l'ordinateur.

### Codage sonore

La durée d'émission sonore ou lumineuse d'un point détermine la vitesse à laquelle le message est envoyé, elle est utilisée en tant que cadence de référence. Voici les conventions de cadence reconnues internationalement :

- Le rythme élémentaire est donné par la durée du point, le *ti*.
- Un *ta* est conventionnellement 3 fois plus long qu'un *ti*.
- L'espacement entre les *ti* et *ta* dans une lettre a la longueur d'un *ti*.
- L'espacement entre les lettres d'un mot a pour longueur un *ta*.
- L'espacement entre les mots est d'au moins 5 *ti*.

Pour produire une tonalité dans la console de *Visual Studio*, vous pouvez exploiter les routines suivantes du fichier d'en-tête

**<windows.h>** :

#### **Beep( tonalité, durée )**

Produit une tonalité sonore via le haut-parleur de l'ordinateur.

*tonalité* : Tonalité du son, en Hertz.

*durée* : Durée du son, en millisecondes.

Exemple : `Beep( 500, 50 );` produit un son de 500 Hz pendant 50 ms. Ce son peut correspondre à un *ti*.

#### **Sleep( durée )**

Suspend temporairement l'exécution du programme.

*durée* : Durée de la pause d'exécution, en millisecondes.

Exemple : `Sleep( 250 );` suspend l'exécution pendant 250 ms. Cette pause peut correspondre à l'espacement entre deux mots (i.e. cinq *ti*).

Pour vous donner un aperçu du résultat attendu de votre part par l'instructeur, celui-ci vous fourni un fichier exécutable de la solution du devoir. Vous pouvez ainsi vous en inspirer pour concevoir votre solution.

### Suggestions

Voici quelques suggestions pour faciliter la réalisation de votre solution à ce devoir :

1. Utilisez l'instruction `std::cin.getline()` pour lire la phrase à traduire en Morse.
2. Définissez une constante `const int TI = 50;` pour représenter la durée d'un *ti*. Vous pouvez ensuite définir les autres constantes

3. Pour avoir plus de détails sur la transmission en code Morse, consultez *Wikipédia* ([http://fr.wikipedia.org/wiki/Alphabet\\_morse](http://fr.wikipedia.org/wiki/Alphabet_morse)), surtout la section intitulée « **Représentation et cadence** »

Ce tableau étant initialisé à la déclaration à l'aides des constantes **TI** et **TA**, vous pouvez y faire référence dans le code afin de convertir les caractères en code Morse sonore.

```
const int TONE = 500;           // fréquence de tonalité sonore

const int VITESSE = 50;        // cadence de référence
                                // vitesse de transmission (en millisecondes)

const int TI = VITESSE;        // durée d'un ti (point) en ms
const int TA = 3 * TI;         // durée d'un ta (trait) en ms

const int ESPACE_SYMBOLE = TI; // durée du silence (en ms) entre chaque ti ou ta d'un caractère
const int ESPACE_LETTRE  = TA; // durée du silence (en ms) entre chaque lettre d'un mot
const int ESPACE_MOT     = 5 * TI; // durée du silence (en ms) entre chaque mot du message
const int MAX_TITA = 6;        // maximum de ti et ta dans un caractère traduisible

// A COMPLETER
const int SIGNAL[][MAX_TITA] = { { TI, TA },          // A
                                  { TA, TI, TI, TI }, // B
                                  . . .
                                  . . .
};

// Fonction de transmission d'un message en Morse
// msg[] : chaîne de caractères à transmettre en Morse sonore
// afficher : true si le code Morse doit aussi être affiché
//            dans la console
// La fonction retourne le nombre de caractères n'ayant pu être transmis

int strMorse( const char msg[], bool afficher ) {

    int compteInconnus = 0;           // compte le nombre de caractères non transmis

    // Traiter chaque caractère du message individuellement
    for ( unsigned int i = 0; i < strlen(msg); i++ ) {
        // Est-ce un espace entre deux mots?
        if ( msg[i] == ' ' ) {
            Sleep( ESPACE_MOT );

            if ( afficher )
                std::cout << ' / ';

            continue;
        }

        // Identifier le caractère et le transmettre
        int idx = -1; // index du caractère dans le tableau signal

        // Les lettres minuscules et majuscule
        // sont traitées sans distinction de case

        // Identifier les caractères MORSES de la table SIGNAL (A COMPLETER)
        . . .
        . . .
        . . .
    }
}
```

```

// Si le caractère est reconnu, alors le transmettre
if ( idx >= 0 )
    // Transmettre chaque symbole du caractère
    for ( int j = 0; j < MAX_TITA; j++ ) {
        if ( SIGNAL[ idx ][ j ] > 0 ) {
            // Transmission visuelle
            if ( afficher )
                if ( SIGNAL[ idx ][ j ] == TI )
                    std::cout << '.';
                else
                    std::cout << '-';

            // Transmission sonore
            Beep( TONE, SIGNAL[ idx ][ j ] );
        }

        // Seulement écouler un délai
        // inter-symboles s'il reste un symbole
        // à transmettre pour le caractère courant
        if ( j < 4 && SIGNAL[ idx ][ j + 1 ] > 0 )
            Sleep( ESPACE_SYMBOLE );
    }

// Écouler le délai inter-lettres avant de passer au prochain caractère
if ( idx >= 0 )
    Sleep( ESPACE_LETTRE );
else {
    // Si le caractère est invalide, l'indiquer visuellement (?) et de
    // façon sonore (pause correspondant à un espace)
    if ( afficher )
        std::cout << ' ';

    Sleep( ESPACE_MOT );

    compteInconnus++;
}

// Retourner le nombre de caractère n'ayant pu être transmis
return compteInconnus;
}

```

**Source:** C++ Comment Programmer, 3<sup>e</sup> édition, Deitel & Deitel, 2001  
Exercice 5.47, p. 386