Ce TD a pour vocation de prendre un peu de recul sur la notion de multimédia que vous utilisez déjà largement. Au lieu de nous intéresser aux usages des outils multimédia (comme vous avez déjà pu le faire au travers d’activités liées au B2i), nous allons nous concentrer sur les aspects plus fondamentaux afin de comprendre comment l’information est stockée dans un ordinateur pour ces différents types de fichiers multimédia.

Ce TD est donc le premier d’une série de 3 sur le multimédia. Nous débutons par le texte ce qui nous permettra de voir comment celui est enregistré dans les fichiers que vous manipulez, puis nous continuerons par le son et enfin les images.

# Introduction générale

Le Multimédia qualifie l'intégration de plusieurs moyens de représentation de l'information, tels que textes, sons, images fixes ou animées.

******

C’est quoi le multimédia ?

## Objets multimédia

Nous appellerons ces différents moyens de représenter de l’information multimédia, des « objets multimédias ». Les objets multimédias de base sont donc : du texte, du son, de l’image et de la vidéo.

Au moins un type de dispositif d’acquisition permet de créer un objet multimédia (ex. un enregistrement sonore) puis au moins un type de dispositif de restitution permet de le rendre accessible à l’utilisateur (ex. par le biais de haut-parleurs). Vous les connaissez bien pour les utiliser fréquemment :

|  |  |  |
| --- | --- | --- |
| Type d’objet multimédia | Dispositif d’acquisition | Dispositif de restitution |
| Du texte | Clavier, ... | Ecran, Imprimante, ... |
| Du son | Microphone, ... | Haut-parleurs, ... |
| De l’image | Appareil photo, scanner, ... | Ecran, imprimante, ... |
| De la vidéo | Caméra, ... | Ecran, haut-parleurs, vidéoprojecteur... |

Tableau : Types de multimédia et dispositifs d'acquisition et de restitution

## Objet multimédia composites

L’intégration de plusieurs objets multimédias de base se fera dans un **objet multimédia** dit **composite**.

Remarquons que la vidéo devrait être considérée comme un objet multimédia composite (images animées et son), même si elle est communément considérée comme un objet multimédia de base.

## Chaîne de traitements multimédia : les grands principes

Une « chaîne de traitements multimédia » est l’ensemble des traitements et modifications possibles sur un objet multimédia entre le dispositif d’acquisition et le dispositif de restitution. Le schéma général d’une chaîne de traitements multimédia peut-être représenté ainsi :



Elle est constituée de différents traitements modifiant l’objet multimédia. Ce dernier peut être sauvegardé dans des fichiers, sous différents formats (format des fichiers avec des extensions qui permet à l’utilisateur de les identifier facilement), à chaque étape (ex. pour le son d’un fichier .wav vers un fichier .mp3).

Voici les détails des différents éléments de la chaîne de traitement :

* Les dispositifs d’acquisition ont été pour la plupart déjà énumérés dans le tableau ci-dessus pour les différents types d’objets multimédias.
* Données avec un codage de base sont celles qui contiennent l’information enregistrée depuis le dispositif d’acquisition sans traitement.
* Les traitements de compression et/ou recodage peuvent avoir plusieurs objectifs. Le premier est de recoder l’information dans les multiples formats plus adaptés aux logiciels qui vont la lire. Le deuxième est de réduire la taille de l’information stockée dans un fichier (rappelez-vous la vocation de zip par exemple) Une fois compresser et/ou recoder, l’information sera enregistrée dans des fichiers avec un encodage différent de celui de base.
* Les fichiers avec un autre encodage sont ceux qui contiennent l’information enregistrée sous différents formats après traitements dans la chaîne.
* Le logiciel d’édition permet alors de lire un fichier multimédia avec un encodage approprié et permet à l’utilisateur de le modifier et de l’enregistrer à nouveau sous forme de fichier.
* Le logiciel de rendu est celui qui va permettre à d’autres utilisateurs de lire le fichier final une fois toutes les modifications de l’utilisateur effectuées sur le logiciel d’édition. Il peut ainsi rendre l’objet multimédia définitif accessible à tout utilisateur (pour le son : par exemple avec Real Player, Winamp …).
* Les dispositifs de restitution ont été pour la plupart déjà énumérés dans le tableau ci-dessus.

Finalement, la majeure partie des logiciels que nous utilisons font partie de ces chaînes de traitement multimédia. Elles sont parfois « insolites ». En faisant preuve d’un peu d’imagination, on peut même en imaginer des nouvelles….

# Chaine de traitements pour le texte

Nous allons dans cette section étudier une chaîne de traitement pour manipuler du texte depuis sa saisie jusqu’à son rendu définitif (ex. son impression). Vous trouverez ci-dessous les spécificités d’une chaîne de traitement pour texte en rouge.



# Codage de texte

Comme vous pouvez vous en douter, les codages de texte sont plus ou moins riches, selon les informations qu’on ajoute au texte pour gérer la mise en forme (ex. dans les fichiers .rtf) et parfois même pour y insérer des images, des feuilles de calculs d’un tableur etc. (ex. dans les fichiers .doc et .docx de Word). Remarquons que dans ce dernier cas, nous manipulons alors un objet multimédia composite…

Et oui tout n’est pas simple … voici donc quelques explications : après le dispositif d’acquisition, ici le clavier, selon les éditeurs de textes, les caractères sont parfois codés différemment. Mais avant d’aller plus loin, essayons de comprendre un peu plus comment tout cela fonctionne.

# Codage de l’information

Pour mieux comprendre, regardons comment l’information est stockée dans un ordinateur.

## Codage de base de l’information dans un ordinateur

L’unité de base pour le stockage de l’information dans un ordinateur est le bit. Dans un bit, on peut stocker une information très simple : soit 1 soit 0 (vrai / faux, oui / non, ying / yang, ...). On parle alors de système binaire.

### Bits et Octets

Pour stocker et représenter une information plus complexe que cette information très simple, on va regrouper les bits par paquets de 8; on appellera un paquet de 8bits, 1 octet (octet vient du latin et du grec signifiant octo, signifiant 8). Avec un octet, on pourra représenter une information plus complexe, avec toutes les possibilités de combinaisons de 0 et de 1 pour chacun des 8 bits. On pourra alors coder par exemple des nombres entre 0 et 255 (soit 256 valeurs = 28). Et si on a besoin de stocker des valeurs encore plus grandes, on fera des paquets de 2 octets (16 bits), 4 octets (32 bits) ou 8 octets (64 bits).

Le symbole utilisé pour l’unité de mesure de l’octet dans le système international d’unités est le « o » minuscule.

### Multiples normalisés de l’octet

Traditionnellement, les préfixes « kilo », « méga », « giga », etc. dans le monde informatique, ne représentaient pas une puissance de 10 (103 = 1 000), mais une puissance de 2 (210 = 1 024). Cependant cette tradition viole les normes en vigueur pour les autres unités, y compris le bit. Une nouvelle norme a donc été créée pour noter les multiples de 210 = 1024 : les « kibi » (kilo binaire), « mébi » (méga binaire), « gibi » (giga binaire), etc. On parlera alors, pour les grandes quantités d’octets, de kibioctet (1 Kio = 210 octets = 1024 octets), mébioctet (1 Mio = 220 octets = 1024 Kio = 1 048 576 octets), gibioctet (1 Gio = 230 octets = 1024 Mo = 1 073 741 824 octets), tebioctet (1 Tio = 240 octets), ...

L'usage traditionnel reste largement en vigueur chez les professionnels comme le grand public, même si c'est en contradiction avec les recommandations qui définissent clairement d'autres préfixes. L'usage des préfixes binaires reste très confidentiel et ne se répand presque pas dans le langage courant, alors que les valeurs représentées par ces unités en puissance de 2 sont très utilisées dans les applications, notamment les systèmes d'exploitation. Cependant, leur utilisation commence à se répandre, notamment dans le monde du logiciel libre, comme dans les systèmes d'exploitation libres de type GNU/Linux.

Cette distinction (entre préfixes binaires et décimaux) est nécessaire, car la confusion entre les deux séries de coefficients est utilisée depuis longtemps par les fabricants de matériels (de disques durs en particulier) pour afficher commercialement des valeurs supérieurs à celles données par les puissances de 2 et ainsi introduire une erreur d’appréciation.

## Base

En arithmétique, un même nombre peut être représenté dans différentes bases. Pour rappel, une base désigne la valeur dont les puissances successives interviennent dans l'écriture des nombres, ces puissances définissant l'ordre de grandeur de chacune des positions occupées par les chiffres composant tout nombre.

Vous manipulez couramment les nombre en base 10. Cela signifie que vous avez 10 symboles (les chiffres de 0 à 9) pour représenter un nombre (10, 123, 9876, ...). Dans le cas d’une notation binaire, on a que deux chiffres pour coder l’information ; 0 et 1 comme on l’a vu précédemment. Donc le nombre 11 (en décimal) se codera en binaire : 00001011 (8 + 2 + 1 = 23 + 21 + 20).

Mais nous pouvons aussi utiliser d’autres bases que la base 10 ou la base 2 pour représenter les nombres. Par exemple, en informatique, on utilise souvent la base 16 (système hexadécimal, facilitant les conversions en base 2 en regroupant des chiffres binaires, 16 étant une puissance de 2). Pour représenter les chiffres au-delà de 9, on utilisera les lettres. Par exemple, pour la base 16, les symboles utilisés sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Vous pouvez utiliser le tableau en annexe (4 premières colonnes) pour prendre quelques exemples de conversions entre nombre décimaux, hexadécimaux ou binaires.

## Théorie de l’information

Et tout cela ne se résume pas simplement à des chiffres et à des conversions. Un des piliers de l’informatique repose sur la théorie de l’information de Shannon[[1]](#footnote-1) qui est une théorie probabiliste permettant de quantifier le contenu moyen en information d'un ensemble de messages, dont le codage informatique satisfait une distribution statistique précise. Cette théorie, communément appelée « théorie de l’information » ou « théorie de la communication », est fondatrice du codage de l’information, de la mesure quantitative de redondance dans un texte, de la compression de données et de la cryptographie.

## Et le codage du texte dans tout cela...

Parmi un grand nombre de codages, voici les plus répandus pour des raisons historiques : l’ASCII étendu intégré dans des codes plus récents et les codes actuellement utilisés dit « UNICODE » comme UTF-8, UTF-16, UTF-32 …

Pour comprendre nous allons ouvrir les fichiers, dans leur format numérique, c’est-à-dire avant que les valeurs des octets du fichier soient converties en caractères. Nous utilisons le logiciel EditHexa que nous installerons depuis l’adresse suivante : <http://stephane.lavirotte.com/teach/cours/envinfo1/EditHexa.zip>.

Vous pouvez retrouver des problèmes de codage en ouvrant des fichiers texte créés sous Windows en ASCII/ANSI avec des éditeurs sous Linux. Ça sera dans ce cas les caractères de saut de ligne qui ne sont pas les mêmes…

Heureusement la plupart des éditeurs gèrent plusieurs encodages en les identifiant, comme vous allez le voir dans les exercices suivants.

# Exercices

## Objets multimédia composites

Voyez-vous d’autres exemples d’objets multimédias composites que ceux cités dans le cours ?

Vidéo et texte (pour les sous-titres), Son et texte (fichier de karaoké), ...

En quoi l’objet « Karaoké » (fichier .kar) est-il un objet multimédia composite ?

Un fichier de karaoké contient du son pour la musique et le texte des paroles que l’on doit chanter. Donc il inclus deux types de base.

## Chaînes de traitements multimédia

Pouvez-vous donner un exemple d’une chaîne de traitement multimédia qui permet d’acquérir du son et de restituer du texte ?

Un système de reconnaissance vocal qui permet soit de commander la machine vocalement, soit de dicter du texte. Exemple de logiciel Dragon Naturally Speaking

Où vient alors s’insérer le traitement spécifique qui se rajoute à la chaîne de traitement multimédia entre le Microphone et l’Ecran par exemple ? Quel est-il ?

Le logiciel de traitement s’insère entre le dispositif d’acquisition (avec le format de données associé) et le logiciel de rendu (le traitement de texte). Donc il prendre la place du logiciel d’édition dans le cadre du schéma présenté.

Pouvez-vous donner un exemple d’une chaîne de traitement multimédia qui permet d’acquérir de l’image et de restituer du texte ?

C’est un logiciel de reconnaissance optique de caractère : on photographe une page, on obtient donc une image et à l’aide d’un logiciel de type OCR (Optical Caractère Recognition), on interprète les données dans l’images pour en extraire les caractères qi y sont représentés.

Où vient alors s’insérer le traitement spécifique qui se rajoute à la chaîne de traitement multimédia entre le Scanner et l’Ecran par exemple ? Quel est-il ?

C’est aussi le logiciel d’édition qui diffère. Un des logiciels fournissant ce type de fonctionnalité sur le marché est Omnipage.

## Bases

1.

De la base 10 vers la base 2. Donner l’écriture en base 2 des nombres suivants :

A = 10(10) B = 31(10)  C = 32(10)

A = 1010(2) B = 11111(2) C = 100000(2)

1.

De la base 2 vers la base 10. Convertir en base 10 les nombres suivants :

D = 101001(2)  E = 10110011(2) F = 1100101(2)  G = 100010111(2)

D = 41(10) E = 179(10) F = 101(10) G = 139(10)

Quel est le nombre le plus grand pour chacun de ces couples de nombres ?

H : 10000(2) et 64(10) I : 10000001(2) et 101(10) J : 10000000(2) 128(10) K : 11111111(2) 256(10)

H : 10000(2)=16(10) < 64(10)

I : 10000001(2)=129(10) > 101(10)

J : 10000000(2)=128(10) = 128(10)

K : 11111111(2)=255(10) 256(10)

## Codage du texte

Nous allons voir que de simples fichiers texte, sans gérer de mise en forme particulière à l’exception du retour à la ligne, peuvent déjà être codés différemment.

1.

Créez un fichier texteANSI.txt sous Windows avec Notepad++ que vous aurez préalablement installé depuis <http://notepad-plus-plus.org/fr/>. Dans ce fichier, mettez le texte « Bonjour ! » et enregistrez-le.

Créez alors deux nouveaux fichiers textes texteUTF8.txt et texteUTF16-BE.txt avec le même texte « Bonjour ! » mais en modifiant le format de codage (cf. menu Encodage, successivement en UTF-8 puis UTF16-BE).

Commencez par démarrer un terminal sous Windows (Démarrer / et taper cmd dans la zone pour rechercher les programmes). Pour afficher successivement le contenu des deux fichiers dans le terminal utilisez la commande type (équivalent de cat sous Unix) :

type nom\_de\_fichier (avec successivement les fichiers texteANSI.txt, texteUTF8.txt puis texteUTF16-BE.txt)

Que remarquez-vous ?

L’affichage des fichiers texteANSI.txt et texteUTF8.txt sont identiques. Par contre l’affiche de texteUTF16-BE.txt affiche bien le même texte, mais avec des espaces entre chaque lettre et un caractère « bizarre » (comme un carré) avant le mot Nonjour !

En comparants les fichiers texteANSI.txt, texteUTF8.txt et texteUTF16-BE.txt, à l’aide de Notepad++, que constatez-vous ?

La taille du deuxième fichier (UTF16-BE.txt) est un peu plus de deux fois plus gros que le fichier textANSI.txt.

En ouvrant les deux fichiers avec le logiciel Bloc-Notes que remarquez-vous ? Comment le logiciel peut connaître l’encodage utilisé par le fichier ?

Quand on ouvre le fichier avec le bloc-note, le logiciel affiche bien le texte tapé. Les premiers octets du fichiers permettent de stocker l’information sur la nature de l’encodage utilisé dans le fichier

Voici par exemple une série de caractère codés en ASCII (ANSI) tels que stockés dans un fichier texte.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6C | 65 | 73 | 20 | 6E | 65 | 75 | 6E | 65 | 75 | 73 | 20 | 66 | 6F | 6E | 74 | 20 | 71 | 75 | 65 |
| 20 | 63 | 6C | 69 | 71 | 75 | 65 | 72 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Utilisez le tableau figurant en annexe pour trouver le texte correspondant.

les neuneus font que cliquer

En UTF16-BE, le codage sera le suivant :

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| FE | FF | 00 | 6C | 00 | 65 | 00 | 73 | 00 | 20 | 00 | 6E | 00 | 65 | 00 | 75 | 00 | 6E | 00 | 65 |
| 00 | 75 | 00 | 73 | 00 | 20 | 00 | 66 | 00 | 6F | 00 | 6E | 00 | 74 | 00 | 20 | 00 | 71 | 00 | 75 |
| 00 | 65 | 00 | 20 | 00 | 63 | 00 | 6C | 00 | 69 | 00 | 97 | 00 | 75 | 00 | 65 | 00 | 72 |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Si l’éditeur de texte le traite comme un fichier avec des caractères codés en ASCII/ANSI, quel sera le texte affiché ?

(Il y a une séparation entre chaque lettre par rapport à la version précédente)

l e s n e u n e u s f o n t q u e c l i q u e r

Cela vous rappelle-t-il quelque chose ?

Le texte est identique (même encodage utilisé en ASCII et en UTF8 pour les caractères dont le code est entre 0 et 255). Comme ici on utilise pas de caractère ayant un vrai code sur 16 bits, le premier octet est toujours à 0. Donc si on interprète les données en ASCII/ANSI, on voit des espace entre chaque lettre (caractère NUL).

Pour aller plus loin…

## Chaînes de traitements multimédia

Décliner un exemple de chaîne de traitement d’objets multimédias, en vous inspirant en particulier du tableau sur les différents types de fichiers multimédia. Si vous connaissez des logiciels qui permettent de travailler sur une ou plusieurs étapes de la chaîne, sous Linux, sous Windows, mentionnez-les.

Chaine de traitement du texte

Dispositif d’acquisition : clavier

Encodage de base : UTF-8

Réencodage : UTF-16

Logiciel de traitement : Notepad, gedit, Word, OpenOffice, ...

Logiciel de rendu: idem + cat, ...

Format de sortie: txt, doc, docx; ...

Dispositif de rendu : écran, imprimante, ...

C’est aussi le logiciel d’édition qui diffère. Un des logiciels fournissant ce type de fonctionnalité sur le marché est Omnipage.

1.

Vers des domaines inexplorés... ou presque … Clavier vers Haut-parleurs ? Image vers Vidéo ? …

Maintenant que vous avez compris, décrivez une chaîne de traitement multimédia de votre invention. Quel traitement spécifique est nécessaire dans cette chaîne de traitement ?

Si l’on conçoit un objet image comme des cases grises (de 0% de noir à 100% de noir) posée sur un damier, et un objet son composé d’une séquence de couples (note, durée), décrivez un algorithme (et oui, c’est ça !) qui permettrait de transformer l’objet image vers un objet son.

Pour toutes les lignes de l’image

 Pour toutes les couples de cases d’une ligne

 Lire la couleur de la case n

 note = Convertir la couleur en un son

 Lire la couleur de la case n+1

 durée = Convertir la couleur en une durée

 Jouer le son(note, durée)

Illustrez cette transformation en traitant deux exemples simples :

* Une image toute grise (que des cases à 50% de noir)

Un seul son qui se qui a toujours la même durée

* Une image « damier » alternant 2 cases noires et 2 cases blanches

Le son joué sera du type deux tons (le pin-pon des pompiers)

## Bases

Dans ces exercices, on passera directement d’une base à l’autre sans passer par la base 10. Pour vous faciliter la conversion, penser à regrouper les bits par 4… et oui, 15(10) = 1111(2) = F(16) !

1. Conversion du binaire vers hexadécimal

Donner l’écriture en base 16 des nombres suivants :

V = 101101(2) W = 101101011110(2) X = 100111001110111(2)

V = 2D(16) W = B5E(16) X = 4E77(16)

1. Conversion de l’hexadécimal vers le binaire

Donner l’écriture en base 2 des nombres suivants :

Y = 24D(16) Z = FE(16)

Y = 1001001101(2) Z = 11111110(2)

# Annexe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Décimal | Octal | Hex | Binaire | Caractère |
| 0 | 0 | 0 | 00000000 | NUL |
| 1 | 1 | 1 | 00000001 | SOH |
| 2 | 2 | 2 | 00000010 | STX |
| 3 | 3 | 3 | 00000011 | ETX |
| 4 | 4 | 4 | 00000100 | EOT |
| 5 | 5 | 5 | 00000101 | ENQ |
| 6 | 6 | 6 | 00000110 | ACK |
| 7 | 7 | 7 | 00000111 | BEL |
| 8 | 10 | 8 | 00001000 | BS |
| 9 | 11 | 9 | 00001001 | HT |
| 10 | 12 | A | 00001010 | LF |
| 11 | 13 | B | 00001011 | VT |
| 12 | 14 | C | 00001100 | FF |
| 13 | 15 | D | 00001101 | CR |
| 14 | 16 | E | 00001110 | SO |
| 15 | 17 | F | 00001111 | SI |
| 16 | 20 | 10 | 00010000 | DLE |
| 17 | 21 | 11 | 00010001 | DC1 |
| 18 | 22 | 12 | 00010010 | DC2 |
| 19 | 23 | 13 | 00010011 | DC3 |
| 20 | 24 | 14 | 00010100 | DC4 |
| 21 | 25 | 15 | 00010101 | NAK |
| 22 | 26 | 16 | 00010110 | SYN |
| 23 | 27 | 17 | 00010111 | ETB |
| 24 | 30 | 18 | 00011000 | CAN |
| 25 | 31 | 19 | 00011001 | EM |
| 26 | 32 | 1A | 00011010 | SUB |
| 27 | 33 | 1B | 00011011 | ESC |
| 28 | 34 | 1C | 00011100 | FS |
| 29 | 35 | 1D | 00011101 | GS |
| 30 | 36 | 1E | 00011110 | RS |
| 31 | 37 | 1F | 00011111 | US |
| 32 | 40 | 20 | 00100000 | SP |
| 33 | 41 | 21 | 00100001 | ! |
| 34 | 42 | 22 | 00100010 | " |
| 35 | 43 | 23 | 00100011 | # |
| 36 | 44 | 24 | 00100100 | $ |
| 37 | 45 | 25 | 00100101 | % |
| 38 | 46 | 26 | 00100110 | & |
| 39 | 47 | 27 | 00100111 | ' |
| 40 | 50 | 28 | 00101000 | ( |
| Décimal | **Octal** | **Hex** | **Binaire** | **Caractère** |
| 41 | 51 | 29 | 00101001 | ) |
| 42 | 52 | 2A | 00101010 | \* |
| 43 | 53 | 2B | 00101011 | + |
| 44 | 54 | 2C | 00101100 | , |
| 45 | 55 | 2D | 00101101 | - |
| 46 | 56 | 2E | 00101110 | . |
| 47 | 57 | 2F | 00101111 | / |
| 48 | 60 | 30 | 00110000 | 0 |
| 49 | 61 | 31 | 00110001 | 1 |
| 50 | 62 | 32 | 00110010 | 2 |
| 51 | 63 | 33 | 00110011 | 3 |
| 52 | 64 | 34 | 00110100 | 4 |
| 53 | 65 | 35 | 00110101 | 5 |
| 54 | 66 | 36 | 00110110 | 6 |
| 55 | 67 | 37 | 00110111 | 7 |
| 56 | 70 | 38 | 00111000 | 8 |
| 57 | 71 | 39 | 00111001 | 9 |
| 58 | 72 | 3A | 00111010 | : |
| 59 | 73 | 3B | 00111011 | ; |
| 60 | 74 | 3C | 00111100 | <  |
| 61 | 75 | 3D | 00111101 | = |
| 62 | 76 | 3E | 00111110 | >  |
| 63 | 77 | 3F | 00111111 | ? |
| 64 | 100 | 40 | 01000000 | @ |
| 65 | 101 | 41 | 01000001 | A |
| 66 | 102 | 42 | 01000010 | B |
| 67 | 103 | 43 | 01000011 | C |
| 68 | 104 | 44 | 01000100 | D |
| 69 | 105 | 45 | 01000101 | E |
| 70 | 106 | 46 | 01000110 | F |
| 71 | 107 | 47 | 01000111 | G |
| 72 | 110 | 48 | 01001000 | H |
| 73 | 111 | 49 | 01001001 | I |
| 74 | 112 | 4A | 01001010 | J |
| 75 | 113 | 4B | 01001011 | K |
| 76 | 114 | 4C | 01001100 | L |
| 77 | 115 | 4D | 01001101 | M |
| 78 | 116 | 4E | 01001110 | N |
| 79 | 117 | 4F | 01001111 | O |
| 80 | 120 | 50 | 01010000 | P |
| 81 | 121 | 51 | 01010001 | Q |
| Décimal | **Octal** | **Hex** | **Binaire** | **Caractère** |
| 82 | 122 | 52 | 01010010 | R |
| 83 | 123 | 53 | 01010011 | S |
| 84 | 124 | 54 | 01010100 | T |
| 85 | 125 | 55 | 01010101 | U |
| 86 | 126 | 56 | 01010110 | V |
| 87 | 127 | 57 | 01010111 | W |
| 88 | 130 | 58 | 01011000 | X |
| 89 | 131 | 59 | 01011001 | Y |
| 90 | 132 | 5A | 01011010 | Z |
| 91 | 133 | 5B | 01011011 | [ |
| 92 | 134 | 5C | 01011100 |  |
| 93 | 135 | 5D | 01011101 | ] |
| 94 | 136 | 5E | 01011110 | ^ |
| 95 | 137 | 5F | 01011111 | \_ |
| 96 | 140 | 60 | 01100000 | ` |
| 97 | 141 | 61 | 01100001 | a |
| 98 | 142 | 62 | 01100010 | b |
| 99 | 143 | 63 | 01100011 | c |
| 100 | 144 | 64 | 01100100 | d |
| 101 | 145 | 65 | 01100101 | e |
| 102 | 146 | 66 | 01100110 | f |
| 103 | 147 | 67 | 01100111 | g |
| 104 | 150 | 68 | 01101000 | h |
| 105 | 151 | 69 | 01101001 | i |
| 106 | 152 | 6A | 01101010 | j |
| 107 | 153 | 6B | 01101011 | k |
| 108 | 154 | 6C | 01101100 | l |
| 109 | 155 | 6D | 01101101 | m |
| 110 | 156 | 6E | 01101110 | n |
| 111 | 157 | 6F | 01101111 | o |
| 112 | 160 | 70 | 01110000 | p |
| 113 | 161 | 71 | 01110001 | q |
| 114 | 162 | 72 | 01110010 | r |
| 115 | 163 | 73 | 01110011 | s |
| 116 | 164 | 74 | 01110100 | t |
| 117 | 165 | 75 | 01110101 | u |
| 118 | 166 | 76 | 01110110 | v |
| 119 | 167 | 77 | 01110111 | w |
| 120 | 170 | 78 | 01111000 | x |
| 121 | 171 | 79 | 01111001 | y |
| 122 | 172 | 7A | 01111010 | z |
| 123 | 173 | 7B | 01111011 | { |
| Décimal | **Octal** | **Hex** | **Binaire** | **Caractère** |
| 124 | 174 | 7C | 01111100 | | |
| 125 | 175 | 7D | 01111101 | } |
| 126 | 176 | 7E | 01111110 | ~ |
| 127 | 177 | 7F | 01111111 | DEL |

1. Shannon C. E., “*A Mathematical Theory of Communications*”, The Bell System Technical Journal, Vol. 27, 1948. [↑](#footnote-ref-1)