Projet PIM 2021-2022

Groupe GH

Format du rapport

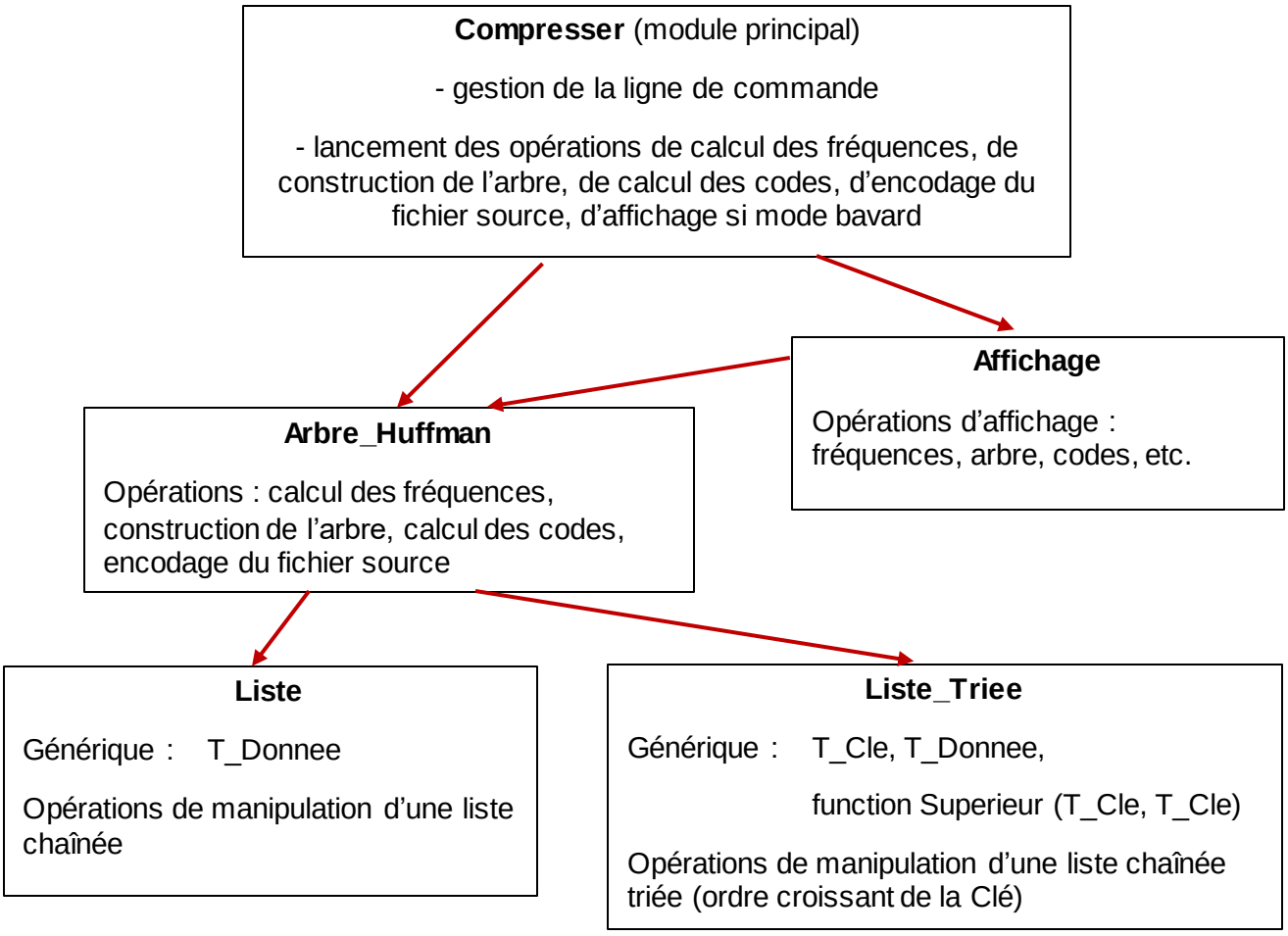
1- Un résumé qui décrit l'objectif et le contenu du rapport (10 lignes maxi)

2- Une introduction qui présente le problème traité

3- Le plan du document

4- L'architecture de l'application en modules

Exemple donné à titre d'illustration :



5- Présentation des principaux choix réalisés

Exemple donné à titre d'illustration :

- Un seul module (Arbre_Huffman) pour implanter les opérations nécessaires de compression et de décompression
- Un module (Affichage) pour gérer tous les affichages, afin de séparer les opération de calcul de l'interface avec l'utilisateur
- Pour gérer la construction de l'arbre, on utilise une liste triée d'arbres (ordre croissant selon la fréquence). La construction de la liste d'arbres (par insertion) se fait en respectant l'ordre, ce qui évite d'avoir à retrier la liste, et permet de récupérer les deux plus petits éléments directement (premiers de la liste).
- Pour stocker le code de huffman d'un smbole, on utilise une liste simple d'octets, chaque octet codant un bit (0 ou 1). Cela permet d'optimiser l'espace mémoire consommé : le code binaire varie de 2 à 16 bits pour le fichier fic_huffman_profond16.txt, même si un tableau peut faire l'affaire sans problème (un peu plus de mémoire contre de meilleures performances d'exécution). Mais il faut prévoir de pouvoir chercher le code courant sans avoir à parcourir la liste depuis le début. Pour cela, on a prévu une fonction qui permet de garder un pointeur sur l'élément courant, et de l'avancer chaque accès.
- Pour stocker les codes de tous les symboles, on utilise un tableau de listes. Le nombre de symboles est connu et est assez faible (256).

6- Principaux algorithmes et types de données

<u>Principaux types de données</u>

```
-- Arbre
type T_Noeud;
type T_Arbre is access T_Noeud;
type T_Noeud is
        record
                Frequence : Integer;
                Donnee : T_Octet;
                fils_gauche : T_Arbre;
                fils_droit : T_Arbre;
        end record;

-- Liste d'arbres
package P_Liste_Triee is
        new Liste_Triee (Integer, T_Arbre, Superieur);
use P_Liste_Triee;
type T_Liste_Arbre is new P_Liste_Triee.T_Liste_Triee;

-- Codes binaires
package P_Liste is
        new Liste (T_Octet);
use P_Liste;
type T_Code_Binaire is new P_Liste.T_Liste;

type T_Codes is array (0..Capacite-1) of T_Code_Binaire;
```

Principaux algorithmes

Construction de l'arbre
Affichage de l'arbre
Construction de la signature de l'arbre
Construction des codes des symboles
Encodage du fichier compressé

```
Encoder le symbole        in : liste_code          -- code binaire du symbole
                          In out : code            -- octet d'encodage
                          In out : nb_bits_remplis --nombre de bits effectifs dans code
Courant <- liste_code
Répéter
        Code <- code * 2                                    -- décalage gauche
        Code := Code + Donnee_Courante (Courant)            -- ajout du bit courant
        Avancer (Courant)                    -- peut être fait dans Donnee_courante
        nb_bits_remplis := nb_bits_remplis + 1
        Si nb_bits_remplis = Taille_Octet Alors
                Ecrire (fichier_hff, code)
                nb_bits_remplis <- 0
                Code <- 0
        Sinon
                Rien
        FinSi
JusquA Est_vide (Courant)
```

-- un éventuel résidu peut se trouver dans code. Il sera traité avec le code du symbole suivant. Au dernier symbole, on effectue Taille_Octet – nb_bits_remplis décalages à gauche.

Reconstruction de l'arbre
Décodage du fichier compressé

7- Démarche adoptée pour tester le programme

8- Difficultés rencontrées et solutions adoptées

9- Organisation de l'équipe (qui a fait quoi, etc.)

10- Bilan technique donnant un état d'avancement du projet (ce qui fonctionne)

Compression complète

$./compresser –b exemple.txt

Frequences:
10 : 2 / 32 : 5 / 58 : 1 / 100 : 1 / 101 : 15 / 108 : 2 / 109 : 4 / 112 : 3 / 116 : 5 / 120 : 4 /

Arbre de Huffman :

(42)
 \--0--(17)
 | \--0--(8)
 | | \--0--(4) 109 m
 | | \--1--(4) 120 x
 | \--1--(9)
 | \--0--(4)
 | | \--0--(2) 10

 | | \--1--(2) 108 l
 | \--1--(5) 32
 \--1--(25)
       \--0--(10)
       | \--0--(5) 116 t
       | \--1--(5)
       | \--0--(2)
       | | \--0--(1) 100 d
       | | \--1--(1)
       | | \--0--(0) 255 ÿ
       | | \--1--(1) 58 :
       | \--1--(3) 112 p
       \--1--(15) 101 e

Code arbre :
7 109 120 10 108 32 116 100 58 112 101 101
0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 1 0 1 1 1 1

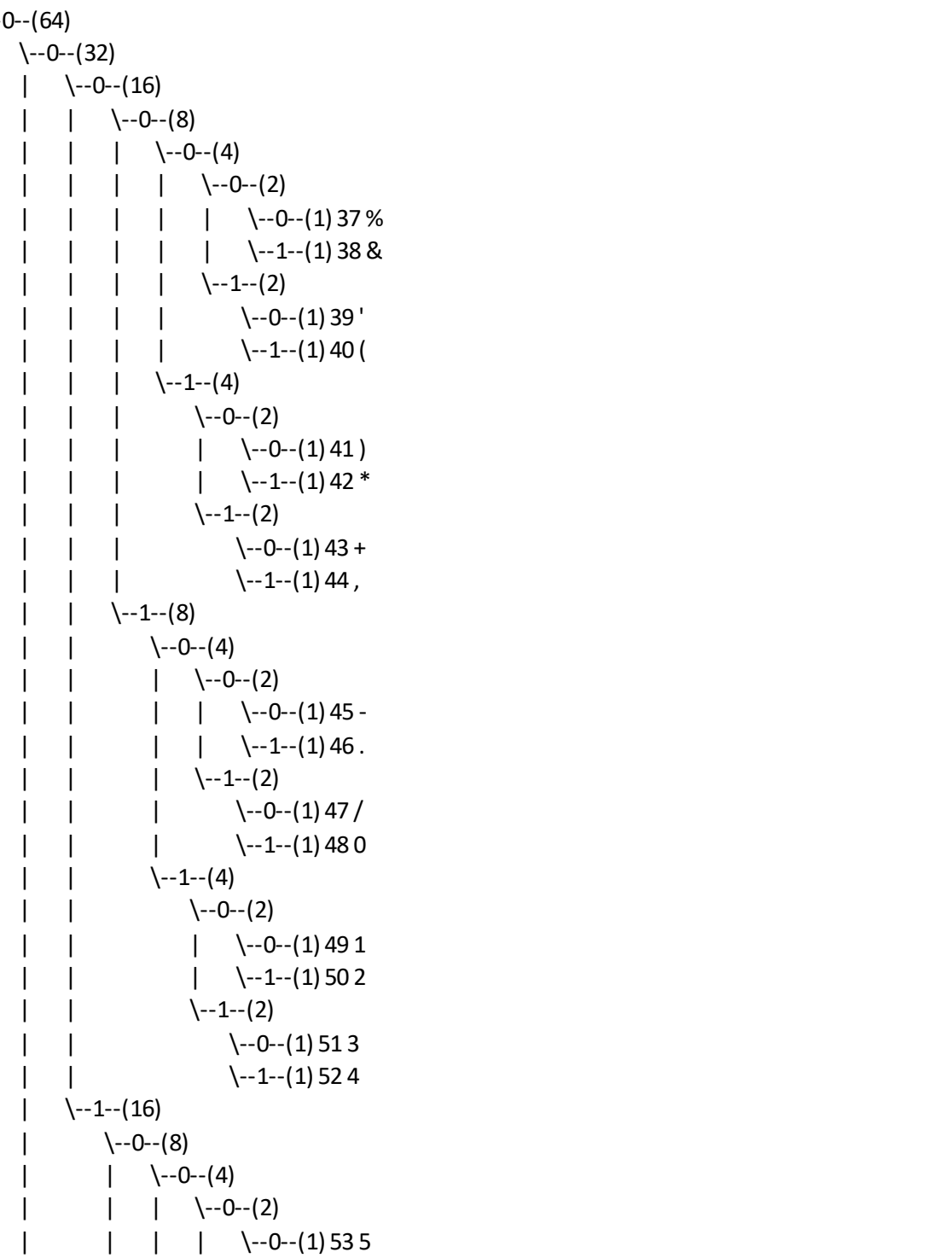
Codes :
109 : 000 / 120 : 001 / 10 : 0100 / 108 : 0101 / 32 : 011 / 116 : 100 / 100 : 10100 / 58 : 101011 / 112 : 1011 / 255 : 101010

$./compresser –b fic_huffman_equilibre.txt

Frequences:
36 : 1 / 37 : 1 / 38 : 1 / 39 : 1 / 40 : 1 / 41 : 1 / 42 : 1 / 43 : 1 / 44 : 1 / 45 : 1 / 46 : 1 / 47 : 1 / 48 : 1 / 49 : 1 / 50 : 1 / 51 : 1 / 52 : 1 / 53 : 1 / 54 : 1 / 55 : 1 / 56 : 1 / 57 : 1 / 58 : 1 / 59 : 1 / 60 : 1 / 61 : 1 / 62 : 1 / 63 : 1 / 64 : 1 / 65 : 1 / 66 : 1 / 67 : 1 / 68 : 1 / 69 : 1 / 70 : 1 / 71 : 1 / 72 : 1 / 73 : 1 / 74 : 1 / 75 : 1 / 76 : 1 / 77 : 1 / 78 : 1 / 79 : 1 / 80 : 1 / 81 : 1 / 82 : 1 / 83 : 1 / 84 : 1 / 85 : 1 / 86 : 1 / 87 : 1 / 88 : 1 / 89

: 1 / 90 : 1 / 91 : 1 / 92 : 1 / 93 : 1 / 94 : 1 / 95 : 1 / 96 : 1 / 97 : 1 / 98 : 1 / 99 : 1 / 100 : 1 / 101 : 1 / 102 : 1 / 103 : 1 / 104 : 1 / 105 : 1 / 106 : 1 / 107 : 1 / 108 : 1 / 109 : 1 / 110 : 1 / 111 : 1 / 112 : 1 / 113 : 1 / 114 : 1 / 115 : 1 / 116 : 1 / 117 : 1 / 118 : 1 / 119 : 1 / 120 : 1 / 121 : 1 / 122 : 1 / 123 : 1 / 124 : 1 / 125 : 1 / 126 : 1 / 127 : 1 / 128 : 1 / 129 : 1 / 130 : 1 / 131 : 1 / 132 : 1 / 133 : 1 / 134 : 1 / 135 : 1 / 136 : 1 / 137 : 1 / 138 : 1 / 139 : 1 / 140 : 1 / 141 : 1 / 142 : 1 / 143 : 1 / 144 : 1 / 145 : 1 / 146 : 1 / 147 : 1 / 148 : 1 / 149 : 1 / 150 : 1 / 151 : 1 / 152 : 1 / 153 : 1 / 154 : 1 / 155 : 1 / 156 : 1 / 157 : 1 / 158 : 1 / 159 : 1 / 160 : 1 / 161 : 1 / 162 : 1 / 163 : 1 /

Arbre de Huffman :



```
|     |          |     |     \--1--(1) 54 6
...
|           |           |           \--0--(1) 79 O
|           |           |           \--1--(1) 80 P
|           |           \--1--(4)
|           |                 \--0--(2)
|           |                 |     \--0--(1) 81 Q
|           |                 |     \--1--(1) 82 R
|           |                 \--1--(2)
|           |                       \--0--(1) 83 S
|           |                       \--1--(1) 84 T
|           \--1--(16)
|                 \--0--(8)
|                       |     \--0--(4)
|                       |     |     \--0--(2)
|                       |     |     |     \--0--(1) 85 U
|                       |     |     |     \--1--(1) 86 V
|                       |     |     \--1--(2)
|                       |     |           \--0--(1) 87 W
|                       |     |           \--1--(1) 88 X
|                       |     \--1--(4)
|                       |           \--0--(2)
|                       |           |     \--0--(1) 89 Y
|                       |           |     \--1--(1) 90 Z
|                       |           \--1--(2)
|                       |                 \--0--(1) 91 [
|                       |                 \--1--(1) 92 \
...
\--1--(64)
      \--0--(32)
      |     \--0--(16)
      |     |     \--0--(8)
      |     |     |     \--0--(4)
      |     |     |     |     \--0--(2)
      |     |     |     |     |     \--0--(1) 101 e
      |     |     |     |     |     \--1--(1) 102 f
      |     |     |     |     \--1--(2)
      |     |     |     |           \--0--(1) 103 g
      |     |     |     |           \--1--(1) 104 h
      |     |     |     \--1--(4)
      |     |     |           \--0--(2)
      |     |     |           |     \--0--(1) 105 i
      |     |     |           |     \--1--(1) 106 j
      |     |     |           \--1--(2)
      |     |     |                 \--0--(1) 107 k
      |     |     |                 \--1--(1) 108 l
      |     |     \--1--(8)
      |     |           \--0--(4)
```

```
|   |       |   \--0--(2)
|   |       |   |   \--0--(1) 109 m
|   |       |   |   \--1--(1) 110 n
|   |       |   \--1--(2)
|   |       |       \--0--(1) 111 o
|   |       |       \--1--(1) 112 p
|   |       \--1--(4)
|   |           \--0--(2)
|   |           |   \--0--(1) 113 q
|   |           |   \--1--(1) 114 r
|   |           \--1--(2)
|   |               \--0--(1) 115 s
|   |               \--1--(1) 116 t
|   \--1--(16)
|       \--0--(8)
|       |   \--0--(4)
|       |   |   \--0--(2)
|       |   |   |   \--0--(1) 117 u
|       |   |   |   \--1--(1) 118 v
|       |   |   \--1--(2)
|       |   |       \--0--(1) 119 w
|       |   |       \--1--(1) 120 x
....
            |   |       \--1--(1) 152 □
            |   \--1--(4)
            |       \--0--(2)
            |       |   \--0--(1) 153 □
            |       |   \--1--(1) 154 □
            |       \--1--(2)
            |           \--0--(1) 155 □
            |           \--1--(1) 156 □
            \--1--(8)
                \--0--(4)
                |   \--0--(2)
                |   |   \--0--(1) 157 □
                |   |   \--1--(1) 158 □
                |   \--1--(2)
                |       \--0--(1) 159 □
                |       \--1--(1) 160
                \--1--(4)
                    \--0--(2)
                    |   \--0--(1) 161 ¡
                    |   \--1--(1) 162 ¢
                    \--1--(2)
                        \--0--(1) 163 £
                        \--1--(1)
                            \--0--(0) 255 ÿ
                            \--1--(1) 36 $
```

Code arbre :
127 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 36 36
0 0 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0 1 1

Codes :
37 : 0000000 / 38 : 0000001 / 39 : 0000010 / 40 : 0000011 / 41 : 0000100 / 42 : 0000101 / 43 : 0000110 / 44 : 0000111 / 45 : 0001000 / 46 : 0001001 / 47 : 0001010 / 48 : 0001011 / 49 : 0001100 / 50 : 0001101 / 51 : 0001110 / 52 : 0001111 / 53 : 0010000 / 54 : 0010001 / 55 : 0010010 / 56 : 0010011 / 57 : 0010100 / 58 : 0010101 / 59 : 0010110 / 60 : 0010111 / 61 : 0011000 / 62 : 0011001 / 63 : 0011010 / 64 : 0011011 / 65 : 0011100 / 66 : 0011101 / 67 : 0011110 / 68 : 0011111 / 69 : 0100000 / 70 : 0100001 / 71 : 0100010 / 72 : 0100011 / 73 : 0100100 / 74 : 0100101 / 75 : 0100110 / 76 : 0100111 / 77 : 0101000 / 78 : 0101001 / 79 : 0101010 / 80 : 0101011 / 81 : 0101100 / 82 : 0101101 / 83 : 0101110 / 84 : 0101111 / 85 : 0110000 / 86 : 0110001 / 87 : 0110010 / 88 : 0110011 / 89 : 0110100 / 90 : 0110101 / 91 : 0110110 / 92 : 0110111 / 93 : 0111000 / 94 : 0111001 / 95 : 0111010 / 96 : 0111011 / 97 : 0111100 / 98 : 0111101 / 99 : 0111110 / 100 : 0111111 / 101 : 1000000 / 102 : 1000001 / 103 : 1000010 / 104 : 1000011 / 105 : 1000100 / 106 : 1000101 / 107 : 1000110 / 108 : 1000111 / 109 : 1001000 / 110 : 1001001 / 111 : 1001010 / 112 : 1001011 / 113 : 1001100 / 114 : 1001101 / 115 : 1001110 / 116 : 1001111 / 117 : 1010000 / 118 : 1010001 / 119 : 1010010 / 120 : 1010011 / 121 : 1010100 / 122 : 1010101 / 123 : 1010110 / 124 : 1010111 / 125 : 1011000 / 126 : 1011001 / 127 : 1011010 / 128 : 1011011 / 129 : 1011100 / 130 : 1011101 / 131 : 1011110 / 132 : 1011111 / 133 : 1100000 / 134 : 1100001 / 135 : 1100010 / 136 : 1100011 / 137 : 1100100 / 138 : 1100101 / 139 : 1100110 / 140 : 1100111 / 141 : 1101000 / 142 : 1101001 / 143 : 1101010 / 144 : 1101011 / 145 : 1101100 / 146 : 1101101 / 147 : 1101110 / 148 : 1101111 / 149 : 1110000 / 150 : 1110001 / 151 : 1110010 / 152 : 1110011 / 153 : 1110100 / 154 : 1110101 / 155 : 1110110 / 156 : 1110111 / 157 : 1111000 / 158 : 1111001 / 159 : 1111010 / 160 : 1111011 / 161 : 1111100 / 162 : 1111101 / 163 : 1111110 / 255 : 11111110

$./compresser –b fic_huffman_profond16.txt

Fréquences:
65 : 1 / 66 : 2 / 67 : 4 / 68 : 8 / 69 : 16 / 70 : 32 / 71 : 64 / 72 : 128 / 73 : 256 / 74 : 512 / 75 : 1024 / 76 : 2048 / 77 : 4096 / 78 : 8192 / 79 : 16384 / 80 : 32768 /

Arbre de Huffman :
(65535)
\--0--(32767)
| \--0--(16383)
| | \--0--(8191)
| | | \--0--(4095)
| | | | \--0--(2047)

```
|    |    |    |    |    \--0--(1023)
|    |    |    |    |    |    \--0--(511)
|    |    |    |    |    |    |    \--0--(255)
|    |    |    |    |    |    |    |    \--0--(127)
|    |    |    |    |    |    |    |    |    \--0--(63)
|    |    |    |    |    |    |    |    |    |    \--0--(31)
|    |    |    |    |    |    |    |    |    |    |    \--0--(15)
|    |    |    |    |    |    |    |    |    |    |    |    \--0--(7)
|    |    |    |    |    |    |    |    |    |    |    |    |    \--0--(3)
|    |    |    |    |    |    |    |    |    |    |    |    |    |    \--0--(1)
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    \--0--(0) 255 ÿ
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    \--1--(1) 65 A
|    |    |    |    |    |    |    |    |    |    |    |    |    |    \--1--(2) 66 B
|    |    |    |    |    |    |    |    |    |    |    |    |    \--1--(4) 67 C
|    |    |    |    |    |    |    |    |    |    |    |    \--1--(8) 68 D
|    |    |    |    |    |    |    |    |    |    |    \--1--(16) 69 E
|    |    |    |    |    |    |    |    |    |    \--1--(32) 70 F
|    |    |    |    |    |    |    |    |    \--1--(64) 71 G
|    |    |    |    |    |    |    |    \--1--(128) 72 H
|    |    |    |    |    |    |    \--1--(256) 73 I
|    |    |    |    |    |    \--1--(512) 74 J
|    |    |    |    |    \--1--(1024) 75 K
|    |    |    |    \--1--(2048) 76 L
|    |    |    \--1--(4096) 77 M
|    |    \--1--(8192) 78 N
|    \--1--(16384) 79 O
 \--1--(32768) 80 P
```

Code arbre :
0 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 80
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Codes :
65 : 0000000000000001 / 66 : 000000000000001 / 67 : 00000000000001 / 68 : 0000000000001 / 69 : 000000000001 / 70 : 00000000001 / 71 : 0000000001 / 72 : 000000001 / 73 : 00000001 / 74 : 0000001 / 75 : 000001 / 76 : 00001 / 77 : 0001 / 78 : 001 / 79 : 01 / 255 : 0000000000000000

Avec valgrind : aucune erreur et aucune perte

```
==176143== Memcheck, a memory error detector
==176143== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==176143== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==176143== Command: ./tcompress fic_huffman_equilibre.txt
==176143==
==176143==
==176143== HEAP SUMMARY:
==176143==     in use at exit: 0 bytes in 0 blocks
==176143==   total heap usage: 277 allocs, 277 frees, 43,692 bytes allocated
```

```
==176143==
==176143==  All heap blocks were freed -- no leaks are possible
==176143==
==176143==  For lists of detected and suppressed errors, rerun with: -s
==176143==  ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Décompression

$./decompresser –b fic_huffman_profond16.txt.hff

Arbre de Huffman :

```
(0)
 \--0--(0)
 |     \--0--(0)
 |     |     \--0--(0)
 |     |     |     \--0--(0)
 |     |     |     |     \--0--(0)
 |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     |     |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     |     |     |     |     |     |     |     \--0--(0)
 |     |     |     |     |     |     |     |     |     |     |     |     |     |     \--0--(0) 255 ÿ
 |     |     |     |     |     |     |     |     |     |     |     |     |     |     \--1--(0) 65 A
 |     |     |     |     |     |     |     |     |     |     |     |     |     \--1--(0) 66 B
 |     |     |     |     |     |     |     |     |     |     |     |     \--1--(0) 67 C
 |     |     |     |     |     |     |     |     |     |     |     \--1--(0) 68 D
 |     |     |     |     |     |     |     |     |     |     \--1--(0) 69 E
 |     |     |     |     |     |     |     |     |     \--1--(0) 70 F
 |     |     |     |     |     |     |     |     \--1--(0) 71 G
 |     |     |     |     |     |     |     \--1--(0) 72 H
 |     |     |     |     |     |     \--1--(0) 73 I
 |     |     |     |     |     \--1--(0) 74 J
 |     |     |     |     \--1--(0) 75 K
 |     |     |     \--1--(0) 76 L
 |     |     \--1--(0) 77 M
 |     \--1--(0) 78 N
 \--1--(0) 79 O
```

Avec valgrind : aucune erreur et aucune perte

```
$valgrind ./decompresser fic_huffman_equilibre.txt.hff
==175316==  Memcheck, a memory error detector
==175316==  Copyright (C) 2002-2017,  and GNU GPL'd, by Julian Seward et al.
==175316==  Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==175316==  Command: ./tdecompress fic_huffman_equilibre.txt.hff
==175316==
==175316==
==175316==  HEAP SUMMARY:
==175316==      in use at exit: 0 bytes in 0 blocks
==175316==    total heap usage: 272 allocs, 272 frees, 34,740 bytes allocated
==175316==
==175316==  All heap blocks were freed -- no leaks are possible
==175316==
==175316==  For lists of detected and suppressed errors, rerun with: -s
==175316==  ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

<u>Vérifications : les fichiers initiaux et les fichiers reconstitués sont identiques</u>

```
$ls –l fic_huffman_profond16.*
-rw-r--r-- 1 hamrouni gea 65535 déc.  18 13:52 fic_huffman_profond16.txt
-rw-r--r-- 1 hamrouni gea 16406 janv.  9 13:19 fic_huffman_profond16.txt.hff
-rw-r--r-- 1 hamrouni gea 65535 janv.  9 13:33 fic_huffman_profond16.txt.hff.dec
$
$diff fic_huffman_profond16.txt fic_huffman_profond16.txt.hff.dec
$
$ls –l fic_huffman_equilibre.*
-rw-r--r-- 1 hamrouni gea 128 déc.  18 13:52 fic_ huffman_ equilibre.txt
-rw-r--r-- 1 hamrouni gea 276 janv.  9 13:39 fic_huffman_ equilibre.txt.hff
-rw-r--r-- 1 hamrouni gea 128 janv.  9 13:43 fic_huffman_ equilibre.txt.hff.dec
$
$diff fic_huffman_ equilibre.txt fic_huffman_ equilibre.txt.hff.dec
$
```

11- Perspectives d'amélioration :

Ce qui peut être amélioré dans un délai raisonnable :
<u>10 jours max, sans modification des modules et des structures de données.</u>

12- Bilan personnel et individuel

13- Modifications faites par rapport aux premiers livrables (structures de données, interfaces des modules, programmes de tests)