

情報通信実験 2(情報理論)課題 1

作成者：Tangjitruboon Tananan

学籍番号：18B09784

1-1 講義資料で挙げた以外の可逆圧縮の規格を 3 つ挙げて主な用途を述べよ。

答え：GIF（画像圧縮形式）,Huffyuv（動画圧縮形式）

1-2 講義資料で挙げた以外の非可逆圧縮の規格を 3 つ挙げて主な用途を述べよ。

答え：MPEG-1（動画圧縮形式）、Vorbis（音声圧縮形式）,WMA（音声圧縮形式）

1-3 講義資料の ASCII コードの符号語長を答えよ。

答え：ASCII コードの符号語長は 8 である。

1-4 情報源アルファベット {a, b, c, d, e} に対する語頭符号の例を 2 つ挙げよ。

答え：

記号	符号語
a	0
b	10
c	110
d	1110
e	11110

記号	符号語
a	000
b	001
c	010
d	011
e	1

1-5 表 1~3 のそれぞれの語頭符号に対して、任意の長さ (n としよう) の記号列 x_n を符号化して符号語列 x_1^n を表示するプログラムを作成して提出せよ。

(1) 【inst enc(a).c】

表 1 語頭符号 1

記号	符号語
a	0
b	1

• 語頭符号 1 : symbols> `bababaaabaaaaaabaabbbbbbabaaa`

答え :

```
[Tananans-MacBook-Pro:exp tananan$ gcc inst_enc\(a\).c
[Tananans-MacBook-Pro:exp tananan$ ./a.out
symbols> bababaaabaaaaaabaabbbbbbabaaa
codewords : 101010001000000010011111101000
```

(2) 【inst enc(b).c】

表 2 語頭符号 2

記号	符号語
a	1
b	01
c	001

• 語頭符号 2 : symbols> `caaaaaacaacaacbaacbbccabbccb`

答え :

```
[Tananans-MacBook-Pro:exp tananan$ gcc inst_enc\(b\).c
[Tananans-MacBook-Pro:exp tananan$ ./a.out
symbols> caaaaaacaacaacbaacbbccabbccb
codewords : 00111111100111100111100101110010101010010011010100100101
```

(3) 【inst enc(c).c】

表 3 語頭符号 3

記号	符号語
a	0
b	10
c	110
d	111

- 語頭符号 3 : symbols>_abcbadcbddbada**cbba**abbac**cb**acdd

答え :

```
[Tananans-MacBook-Pro:exp tananan$ gcc inst_enc\(c\).c
[Tananans-MacBook-Pro:exp tananan$ ./a.out
symbols> abcbadcbddbadacbbaabbaccbacdd
codewords : 0101100100111110101111111001110110101000101001101101001101111111
```

1-6 表 1~3 の語頭符号に対して、任意の長さの符号語列を復号化して記号列を表示するプログラムを作成して提出せよ。

(1) 【inst dec(a).c】

- 語頭符号 1 : codewords>_1111110100010011111111111101001

答え :

```
[Tananans-MacBook-Pro:exp tananan$ gcc inst_dec\(a\).c
[Tananans-MacBook-Pro:exp tananan$ ./a.out
codewords > 1111110100010011111111111101001
symbols : bbbbbbbabaaabaabbbbbbbbbbbabab
```

(2) 【inst dec(b).c】

- 語頭符号 2 : codewords>_0011100110101001100110010011001

答え :

```
[Tananans-MacBook-Pro:exp tananan$ gcc inst_dec\(b\).c
[Tananans-MacBook-Pro:exp tananan$ ./a.out
codewords > 0011100110101001100110010011001
symbols : caacabbacacaccac
```

(3) 【inst dec(c).c】

- 語頭符号 3 : codewords> 1011110110111111100011011010

答え：

```
[Tananans-MacBook-Pro:exp tananan$ gcc inst_dec\c\c
[Tananans-MacBook-Pro:exp tananan$ ./a.out
codewords > 1011110110111111100011011010
symbols : bdbcd dbaac cb
```

1-7 【kraft.c】クラフトの不等式を満たす符号語長列に対する語頭符号を表示するプログラムを作成して提出せよ。

- 次の入力に対する実行結果を答えよ。

```
alphabet size> 6
l_1> 1
l_2> 2
l_3> 4
l_4> 8
l_5> 16
l_6> 32
```

答え：

```
[Tananans-MacBook-Pro:exp tananan$ gcc kraft.c
[Tananans-MacBook-Pro:exp tananan$ ./a.out
alphabet size> 6
l_1> 1
l_2> 2
l_3> 4
l_4> 8
l_5> 16
l_6> 32
cw for l_1: 0
cw for l_2: 10
cw for l_3: 1100
cw for l_4: 11010000
cw for l_5: 1101000100000000
cw for l_6: 11010001000000010000000000000000
```

1-8 【shannon.c】与えられた確率分布 $\{p_i\}_{i=1}^M$ に対する Shannon 符号を表示するプログラムを作成して提出せよ。また、与えられた確率分布に対するエントロピーと、Shannon 符号の平均符号語長も出力すること。

- 次の入力に対する実行結果を答えよ。

alphabet_size> 8

p_1> 0.261

p_2> 0.241

p_3> 0.152

p_4> 0.131

p_5> 0.115

p_6> 0.064

p_7> 0.034

p_8> 0.002

答え：

```
[Tananans-MacBook-Pro:exp tananan$ gcc shannon.c
```

```
[Tananans-MacBook-Pro:exp tananan$ ./a.out
```

```
alphabet size> 8
```

```
p_1> 0.261
```

```
p_2> 0.241
```

```
p_3> 0.152
```

```
p_4> 0.131
```

```
p_5> 0.115
```

```
p_6> 0.064
```

```
p_7> 0.034
```

```
p_8> 0.002
```

```
cw for l_1: 00
```

```
cw for l_2: 010
```

```
cw for l_3: 011
```

```
cw for l_4: 100
```

```
cw for l_5: 1010
```

```
cw for l_6: 1011
```

```
cw for l_7: 11000
```

```
cw for l_8: 110010000
```

```
entropy: 2.594226
```

```
average length: 2.998000
```

1-10 【not inst enc.c】 表 4 の符号（語頭符号ではない）に対して、任意の長さの符号語列を復号化して記号列を表示する プログラムを作成して提出せよ。

表 4 符号

記号	符号語
a	10
b	00
c	11
d	110

```
codewords> 10001101011101101000110101000110101110110100011010
```

答え：

```
[Tananans-MacBook-Pro:exp tananan$ gcc not_inst_enc.c
[Tananans-MacBook-Pro:exp tananan$ ./a.out
codewords > 10001101011101101000110101000110101110110100011010
symbols : abdacadabdaabdacadabda
```

1-11 【given inst enc.cpp】 任意に与えられた語頭符号に対して、任意の長さの記号列を符号化して符号語列を表示するプログラムを作成して提出せよ。

• 次の入力に対する実行結果を答えよ。

```
alphabet_size> 4
symbol_0> a
codeword_0> 0011
symbol_1> b
codeword_1> 01110
symbol_2> c
codeword_2> 1
symbol_3> d
codeword_3> 0010
symbols> abcabadcbddbadaacbbabbaccbacdd
```

答え：

```
[Tananans-MacBook-Pro:exp tananan$ python3 given_inst_enc.py
alphabet size> 4
symbol_1> a
codeword_1> 0011
symbol_2> b
codeword_2> 01110
symbol_3> c
codeword_3> 1
symbol_4> d
codeword_4> 0010
symbols> abcabadcbddbadaacbbabbaccbacdd
codewords: 00110111010011011100011001010111000100010011100011001000111
0111001110001100110111001110001111011100011100100010
_
```

1-12 【given inst dec.cpp】 任意に与えられた語頭符号に対して、任意の長さの符号語列を復号化して記号列を表示するプログラムを作成して提出せよ。

- 次の入力に対する実行結果を答えよ。

```
alphabet_size> 4
symbol_0> a
codeword_0> 0011
symbol_1> b
codeword_1> 01110
symbol_2> c
codeword_2> 1
symbol_3> d
codeword_3> 0010
codewords> 0011011100011001100110010011100010001010011011101001101110100
101001100111100110011101110001100100111001110
```

答え：

```
[Tananans-MacBook-Pro:exp tananan$ python3 given_inst_dec.py ]
alphabet size> 4
symbol_1> a
codeword_1> 0011
symbol_2> b
codeword_2> 01110
symbol_3> c
codeword_3> 1
symbol_4> d
codeword_4> 0010
codeword> 0011011100011001100110010011100010001010011011101001101110100101001100
111100110011101110001100100111001110
symbols: abaaadbddcabcbcdcaaccaachadbb
```


情報通信実験 2(情報理論)課題 2

作成者：Tangjitruamboon Tananan

学籍番号：18B09784

2-1 【iid.c】 与えられた 1 記号の生起確率 $\{p_i\}_{i=1}^M$ に従う無記憶情報源について、記号列の生起確率を出力する プログラムを作成して提出せよ。

- 次の入力に対する実行結果を答えよ。

alphabet size> 3

length> 4

symbol_1> a

p_1> 0.1

symbol_2> b

p_2> 0.2

symbol_3> c

p_3> 0.7

答え：

```
Tananans-MacBook-Pro:exp tananan$ python3 iid.py
alphabet size> 3
length> 4
symbol_1> a
p_1> 0.1
symbol_2> b
p_2> 0.2
symbol_3> c
p_3> 0.7
P(aaab):0.0002
P(aaac):0.0007
P(aaaa):0.0001
P(aaba):0.0002
P(aabb):0.0004
P(aabc):0.0014
P(aacb):0.0014
P(aacc):0.0049
P(aaca):0.0007
P(abba):0.0004
P(abbb):0.0008
P(abbc):0.0028
P(abcb):0.0028
P(abcc):0.0098
P(abca):0.0014
P(abaa):0.0002
P(abab):0.0004
P(abac):0.0014
P(acab):0.0014
P(acac):0.0049
P(aca):0.0007
P(acba):0.0014
P(acbb):0.0028
P(acbc):0.0098
P(accb):0.0098
P(acco):0.0343
P(acca):0.0049
P(bbba):0.0008
P(bbbb):0.0016
P(bbbc):0.0056
P(bbb):0.0056
P(bbcc):0.0196
P(bbca):0.0028
P(bbaa):0.0004
P(bbab):0.0008
```

```
P(bbca):0.0028
P(bbaa):0.0004
P(bbab):0.0008
P(bbac):0.0028
P(bcab):0.0028
P(bcac):0.0098
P(bcaa):0.0014
P(bcba):0.0028
P(bcb):0.0056
P(bcbc):0.0196
P(bccb):0.0196
P(bccc):0.0686
P(bcca):0.0098
P(baba):0.0004
P(babb):0.0008
P(babc):0.0028
P(bacb):0.0028
P(bacc):0.0098
P(baca):0.0014
P(baaa):0.0002
P(baab):0.0004
P(baac):0.0014
P(caab):0.0014
P(caac):0.0049
P(caaa):0.0007
P(caba):0.0014
P(cabb):0.0028
P(cabc):0.0098
P(cacb):0.0098
P(cacc):0.0343
P(caca):0.0049
P(cbba):0.0028
P(cbbb):0.0056
P(cbbc):0.0196
P(cbc):0.0196
P(cbcc):0.0686
P(cbca):0.0098
P(cbaa):0.0014
P(cbab):0.0028
P(cbac):0.0098
P(ccab):0.0098
P(ccac):0.0343
P(ccaa):0.0049
P(ccba):0.0098
P(ccbb):0.0196
P(ccbc):0.0686
P(cccb):0.0686
P(cccc):0.2401
P(ccca):0.0343
```


2-3 算術符号の復号では記号列長を知っていることを仮定している。これを実現するには例えば、ガンマ符号と呼ばれる符号を利用すればよい。ガンマ符号をどのように利用すればよいだろうか？ ガンマ符号の説明とともに、その具体的な利用方法を説明せよ。

答え：

ガンマ符号はビットレベルでの整数の符号化手法である。ガンマ符号は整数 x を $1 + \text{floor}(\log_2 x)$ を unary 符号で、 $x - 2^{\text{floor}(\log_2 x)}$ をバイナリ表現した $\text{floor}(\log_2 x)$ ビットとする符号化手法である。

例えば、 $x=6$ の場合 $\text{floor}(\log_2 6) + 1 = 2 + 1 = 3$ unary 符号で符号化でバイナリ部分が 2 ビットを示し、110になる。また、 $6 - 2^{\text{floor}(\log_2 6)} = 6 - 2^2 = 2$ ビットで表現し、10 できる。そのため、6 のガンマ符号は110 10 とわかる。

このような考え方で、10のガンマ符号は11000 010ができ、ガンマ符号は8 ビット以下で表現できていることが分かります。

一般的な文書コーパスを対象に転置インデックスを構築した際は Variable Byte Code よりも γ 符号で文書 ID を符号化の方が全体として圧縮率は高い。RCV1 の転置インデックス は 32 ビット固定のバイナリ符号で 400MB になるところ、Variable Byte Code で 116 MB、ガンマ符号で 101 MB になる。

2-4 【arithmetic code.c】算術符号を実装するプログラムを作成して提出せよ。

- 次の入力に対する実行結果を答えよ。

```
alphabet size> 4
symbol_1> a
p_1> 0.6
symbol_2> b
p_2> 0.2
symbol_3> c
p_3> 0.1
symbol_4> d
p_4> 0.1
symbols> aadaadaabcccdbcbacc
codewords> 00101011001100110011
length> 30
```

答え：

```
[Tananans-MacBook-Pro:exp tananan$ python3 arithmetic_code.py
alphabet size> 4
symbol_1> a
p_1> 0.6
symbol_2> b
p_2> 0.2
symbol_3> c
p_3> 0.1
symbol_4> d
p_4> 0.1
symbols> aadaadaabcccdbcbacc
range: [0.336026871284,0.336026871284)
encoded: 0101011000000101110110110110011010111110000000
codewords> 00101011001100110011
length> 30
decode: aaabdaaaaacaaaaabababcbbbbadd
```

2-6 【markov.c】 与えられた 1 記号の生起確率と、1 記号が生起した後に 1 記号が生起する確率（条件付き確率）に従うマルコフ情報源について、記号列の生起確率を出力するプログラムを作成して提出せよ。

• 次の入力に対する実行結果を答えよ。

```
alphabet size> 2
length> 5
symbol_1> a
p_1> 0.2
symbol_2> b
p_2> 0.8
cp_11> 0.2
cp_12> 0.8
cp_21> 0.5
cp_22> 0.5
```

答え：

```
[Tanananans-MacBook-Pro:exp tananan$ python3 markov.py
alphabet size> 2
length> 5
symbol_1> a
p_1> 0.2
symbol_2> b
p_2> 0.8
cp_11> 0.2
cp_12> 0.8
cp_21> 0.5
cp_22> 0.5
P(aaaaa):0.00032
P(aaaab):0.00128
P(aaabb):0.0032
P(aaaba):0.0032
P(aabba):0.008
P(aabbb):0.008
P(aabab):0.0128
P(aabaa):0.0032
P(abbba):0.008
P(abbab):0.032
P(abbbb):0.02
P(abbba):0.02
P(ababa):0.032
P(ababb):0.032
P(abaab):0.0128
P(abaaa):0.0032
P(bbaaa):0.008
P(bbaab):0.032
P(bbabb):0.08
P(bbaba):0.08
P(bbbba):0.05
P(bbbbb):0.05
P(bbbab):0.08
P(bbbba):0.02
P(babaa):0.032
P(babab):0.128
P(babbb):0.08
P(babba):0.08
P(baaba):0.032
P(baabb):0.032
P(baaab):0.0128
P(baaaa):0.0032
```