問題：半角英数字、全角カナ、全角漢字、半角バックスラッシュ(ASCII)、半角円記号 (JIS X 0201) 等、色々な文字を含むテキストファイルを観察する。

1. 準備したデータ

   使用するデータは JIS X 0208 符号化文字集合 (JIS 漢字) に含まれる全角文字と、ASCII、JIS に含まれる半角文字を UTF-8 方式で符号化したファイル (x0208UTF8.txt) とする。x0208UTF8.txt の内容を次に示す。

```
$ cat x0208UTF8.txt
0Aa!
あア亜院
a\￥A
$
```

2. UTF-8

   UTF-8 に符号化した状態を 16 進ダンプして内容を確認する。

```
$ hexdump x0208UTF8.txt     # もともと UTF-8 だからそのまま 16 進ダンプ
0000000 30 41 61 21 0a e3 81 82 e3 82 a2 e4 ba 9c e9 99
0000010 a2 0a 61 5c c2 a5 41 0a
0000018
$

// UTF-8 に変換した結果を解析すると次のようになっている
30       : U+0030 : BMP (ASCII 領域) 0
41       : U+0041 : BMP (ASCII 領域) A
61       : U+0061 : BMP (ASCII 領域) a
21       : U+0021 : BMP (ASCII 領域) !
0a       : U+000a : BMP (ASCII 領域) LF
e3 81 82 : 11100011 10000001 10000010 (2 進数で書いた)
        => ----0011 --000001 --000010 (データ表しているビットを抜き出す)
        => 00110000 01000010          (データ表しているビットだけ並べる)
        => 30 42 = U+3042 = あ        (16 進数に変換すると Unicode)
e3 82 a2 : 11100011 10000010 10100010 (2 進数で書いた)
        => ----0011 --000010 --100010 (データ表しているビットを抜き出す)
        => 00110000 10100010          (データ表しているビットだけ並べる)
        => 30 a2 = U+30a2 = ア        (16 進数に変換すると Unicode)
e4 ba 9c : 11100100 10111010 10011100 (2 進数で書いた)
        => ----0100 --111010 --011100 (データ表しているビットを抜き出す)
        => 01001110 10011100          (データ表しているビットだけ並べる)
        => 4e 9c = U+4e9c = 亜        (16 進数に変換すると Unicode)
e9 99 a2 : 11101001 10011001 10100010 (2 進数で書いた)
        => ----1001 --011001 --100010 (データ表しているビットを抜き出す)
        => 10010110 01100010          (データ表しているビットだけ並べる)
        => 96 62 = U+9662 = 院        (16 進数に変換すると Unicode)
0a       : U+000a : BMP (ASCII 領域) LF
61       : U+0061 : BMP (ASCII 領域) a
5c       : U+005c : BMP (ASCII 領域) \
c2 a5    : 11000010 10100101          (2 進数で書いた)
        => ---00010 --100101          (データ表しているビットを抜き出す)
        => 000 10100101               (データ表しているビットだけ並べる)
        => 00 a5 = U+00a5 =  ￥       (16 進数に変換すると Unicode)
41       : U+0041 : BMP (ASCII 領域) A
0a       : U+000a : BMP (ASCII 領域) LF
```

3. ISO-2022-JP

   ISO-2022-JP に符号化した状態を 16 進ダンプして内容を確認する。

```
$ iconv -f UTF-8 -t ISO-2022-JP x0208UTF8.txt | hexdump
0000000 30 41 61 21 0a 1b 24 42 24 22 25 22 30 21 31 21
0000010 1b 28 42 0a 61 5c 1b 28 4a 5c 1b 28 42 41 0a
000001f
$

// ISO-2022-JP に変換した結果を解析すると次のようになっている
30       : ASCII コード表 (30h)        : 0
41       : ASCII コード表 (41h)        : A
61       : ASCII コード表 (61h)        : a
21       : ASCII コード表 (21h)        : !
0a       : ASCII コード表 (0ah)        : LF
1b 24 42 : ESC $ B                    : JIS X 0208 (JIS 漢字) に切り替える
24 22    : JIS X 0208 コード表 (2422h) : あ
25 22    : JIS X 0208 コード表 (2522h) : ア
30 21    : JIS X 0208 コード表 (3021h) : 亜
31 21    : JIS X 0208 コード表 (3121h) : 院
1b 28 42 : ESC ( B                    : ASCII に切り替える
0a       : ASCII コード表 (0ah)        : LF
61       : ASCII コード表 (61h)        : a
5c       : ASCII コード表 (5ch)        : \
1b 28 4a : ESC ( J                    : JIS X 0201 (JIS 8bit コード) に切り替える
5c       : JIS X 0201 コード表 (5ch)   : ￥
1b 28 42 : ESC ( B                    : ASCII に切り替える
41       : ASCII コード表 (41h)        : A
0a       : ASCII コード表 (0ah)        : LF
```

4. Shift_JIS

   Shift_JIS に符号化した状態を 16 進ダンプして内容を確認する。

```
$ iconv -f UTF-8 -t SJIS x0208UTF8.txt | hexdump

iconv: x0208UTF8.txt:3:1: cannot convert                <--- エラーが発生！！
0000000 30 41 61 21 0a 82 a0 83 41 88 9f 89 40 0a 61
000000f
$

// Shift_JIS に変換した結果を解析すると次のようになっている
30       : ASCII コード表 (30h)        : 0
41       : ASCII コード表 (41h)        : A
61       : ASCII コード表 (61h)        : a
21       : ASCII コード表 (21h)        : !
0a       : ASCII コード表 (0ah)        : LF
82 a0    : JIS 漢字コード表 (2422h)    : あ
83 41    : JIS 漢字コード表 (2522h)    : ア
88 9f    : JIS 漢字コード表 (3021h)    : 亜
89 40    : JIS 漢字コード表 (3121h)    : 院
0a       : ASCII コード表 (0ah)        : LF
61       : ASCII コード表 (61h)        : a

ここまででエラー終了('\' を Shift_JIS に変換できない)
```

5. EUC-JP

   EUC-JP に符号化した状態を 16 進ダンプして内容を確認する。

```
$ iconv -f UTF-8 -t EUC-JP x0208UTF8.txt | hexdump
0000000 30 41 61 21 0a a4 a2 a5 a2 b0 a1 b1 a1 0a 61 5c
0000010 5c 41 0a
0000013
$

// EUC-JP に変換した結果を解析すると次のようになっている
30       : ASCII コード表 (30h) : 0
41       : ASCII コード表 (41h) : A
61       : ASCII コード表 (61h) : a
21       : ASCII コード表 (21h) : !
0a       : ASCII コード表 (0ah) : LF
a4 a2    : JIS 漢字コード表 (a4a2h - 8080h = 2422h) : あ
a5 a2    : JIS 漢字コード表 (a5a2h - 8080h = 2522h) : ア
b0 a1    : JIS 漢字コード表 (b0a1h - 8080h = 3021h) : 亜
b1 a1    : JIS 漢字コード表 (b1a1h - 8080h = 3121h) : 院
0a       : ASCII コード表 (0ah) : LF
61       : ASCII コード表 (61h) : a
5c       : ASCII コード表 (5ch) : \
5c       : ASCII コード表 (5ch) : \        <- ￥だったはずが！！
41       : ASCII コード表 (41h) : A
0a       : ASCII コード表 (0ah) : LF
```

6. EUC-JP に変換した後 UTF-8 に戻す

   UTF-8 から EUC-JP に変換した後、EUC-JP から UTF-8 に逆変換する。正しく変換できるなら変換前と同じものに戻るはずだが、元に戻らない。

```
$ cat x0208UTF8.txt
0Aa!
あア亜院
a\￥A
$ iconv -f UTF-8 -t EUC-JP x0208UTF8.txt | iconv -f EUC-JP -t UTF-8
0Aa!
あア亜院
a\\A                        <--- もとの "a\￥A" と異なる！！(元に戻らない)
$
```