文字コード

1. はじめに

この演習では，計算機で文字をどのように扱っているかについて学びます．計算機では文字を数値で表現しています．たとえば，ある計算機ではアルファベットの A を 41h（h は 16 進数表記を意味することとします）という数値で表現します．ある計算機と書いたのは，計算機によってどのような数値がどのような文字を表現しているかは違うことがあるからです．この決まり，どのような数値がどのような文字を表現するかという決まりの集合が文字コードです．

この演習の目的は，現在使われている文字コードを知り，その扱い方の違いを理解して，実際にプログラムの中で使えるようになることです．

2. 文字コードとは

先にも書いたように，文字コードとはなんという文字がどの数値（コードポイント）で表現されているかという決まりの集合です．文字コードは暗号のようなものでもあります．ある数値がどの文字に対応するかは文字コードによって異なり，どのような文字コードを使っているかを知らないと何が記されているのかわからないからです．つまり，文字コードを勝手に決めてしまうと他の計算機とデータの交換ができなくなってしまいます．そこで，文字コードにも標準規格を作ろうという動きが起こり，現在使用されている ASCII コードや JIS コードなどが生まれました．

ここで，ASCII コードの正式名は American national Standard Code for Information Interchange です．また JIS コードは情報交換用符号と呼びます．この名前からこれらは情報を交換することを目的とした文字コードであることがわかります．つまり，交換するときの形式を決めておけば混乱なく情報を交換できるという考えに基づいて作られたということです．ただし，処理を簡単にするため，これら情報交換用の文字コードをそのまま計算機の内部コードとしても使っています．

文字コード（符号化文字集合）は，文字セット（文字集合）と符号化方式から定義されます（図１）．文字セットとは扱う文字の集合です．一方の符号化方式とはどのように数値に対応させるかというルールです．文字コードを使うときには，どのような文字セットと符号化方式を用いているかが重要になります．文字セットの特徴はどのような文字を含んでいるかそのものです．単純にどの文字を含めているかという点に加え，字体（明朝体やゴシック体）を別々の文字としているか，異字体（高と髙）を別々の文字としているかという点があります．符号化方式の特徴としては順序関係と連続性が

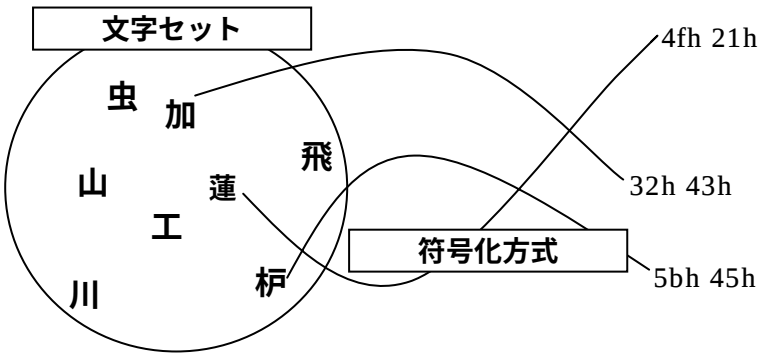

図１　文字コードの定義要素

挙げられます．アルファベットであれば a,b,c,…，日本語であれば 50 音順に並んでいるかという順序はコードポイントを用いて整列処理（ソート）を行うときに重要です．また，A のコードポイントと B のコードポイントが連続であるかどうかも利用する際に重要な点となります．

3. いろいろな文字コード

3.1 ASCII コード

ASCII コードは ANSI（日本の JIS にあたる）規格の文字コードです．ASCII コードは 7 桁の 2 進数（7 ビット）で文字を表現し，0～7fh(127)をコードポイントに用いて，128 種類の文字を表現可能です．ASCII コード表を表 1 に示します．行が上位 3 ビット，列が下位 4 ビットを示しますので，たとえば A のコードポイントは 41h となります．

表１ ASCII **コード表**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

この ASCII コードが元となり，国際標準化機構 ISO の規格 ISO 646 が誕生しました．ASCII コードとの違いは，いくつかのコードポイントは各国で自由に文字を割り振ってよい点です．米国マイクロソフト社の Windows などにおいて，日本語モードでは ¥ と表示される部分が英語モードでは \ となるのがこの例です．

そして，アルファベットや数字に加えカタカナを文字セットに含む文字コードとして，日本で作られたのが ISO 646 準拠の JIS X 0201（7 ビット及び 8 ビットの情報交換用符号化文字集合）です．JIS X 0201 には 7 ビット系（JIS7）と 8 ビット系（JIS8）の二つが含まれています（図２）．JIS7 は ISO 646 の自由定義部分を日本用にしたもの（JIS ローマ字）と，ISO 646 の制御文字の後ろにカタカナを割り当てたもの（JIS カタカナ：表２）の二つから構成され，この二つを制御文字で切り替えて使用します．JIS8 は 0～7fh(127)番までに JIS ローマ字を，80h(128)～ffh(255)番までに JIS カタカナを配置したものです．

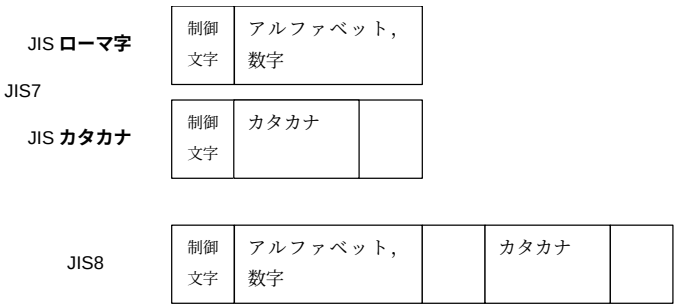

図２ JIS X 0201 **の概念図**

表２ JIS **カタカナ**（JIS X0201 **のカタカナ部分**）**コード表**

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC		「	」			ヲ	ァ	ィ	ゥ	ェ	ォ	ャ	ュ	ョ	ッ
3	-	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	サ	シ	ス	セ	ソ
4	タ	チ	ツ	テ	ト	ナ	ニ	ヌ	ネ	ノ	ハ	ヒ	フ	ヘ	ホ	マ
5	ミ	ム	メ	モ	ヤ	ユ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ン	"	°
6																
7																DEL

3.2 **漢字文字セット**

漢字を扱う文字コードの説明の前に漢字を含む文字セットについて説明します．漢字を含む文字セットの標準となっている JIS 規格は JIS X 0208（７ビット及び８ビットの２バイト情報交換用符号化漢字集合）です．次に説明する JIS コード，シフト JIS，EUC はすべてこの文字セットを表現しています．JIS X 0208 には非漢字 453 字，第一水準 2965 字，第二水準の合計 6879 字が含まれています．また，これに含まれていない漢字 6067 文字を含む補助漢字文字セット（JIS X 0212）というものもありますが，現在これを使える計算機は多くありません．さらに 2000 年 1 月には JIS X 0208 に第三水準，第四水準が加わった JIS X 0213 が制定されました．この文字セットの注目すべき点は，いままで機種依存であった①のような文字が規格に入った点です．

3.3 JIS **コード**

いわゆる JIS コードと呼ばれているのは，ISO 2022 と呼ばれる文字コードを切り替えるための方法の規定に従って，JIS X 0208 の文字セットを 7 ビットの番号二つの組み合わせ（２バイト）で文字を表現した ISO-2022-JP です．たとえば“加”という文字のコードポイントは 32h 43h という二つの番号の組となります．ISO と付きますが ISO の規格ではありません．また，JIS コードが通称ですが JIS の規格でもありません．

JIS X 0208 の文字セットは漢字です．アルファベットや数字をあらわす場合は ASCII や JIS ローマ字を利用します．これらの文字コードと共存させるために，使っている番号は 7 ビットの範囲で ASCII の制御文字の部分を除いた 20h～7Fh だけです．したがって，表現できる文字種は 94×94=8836 種だけとなります（図３）．

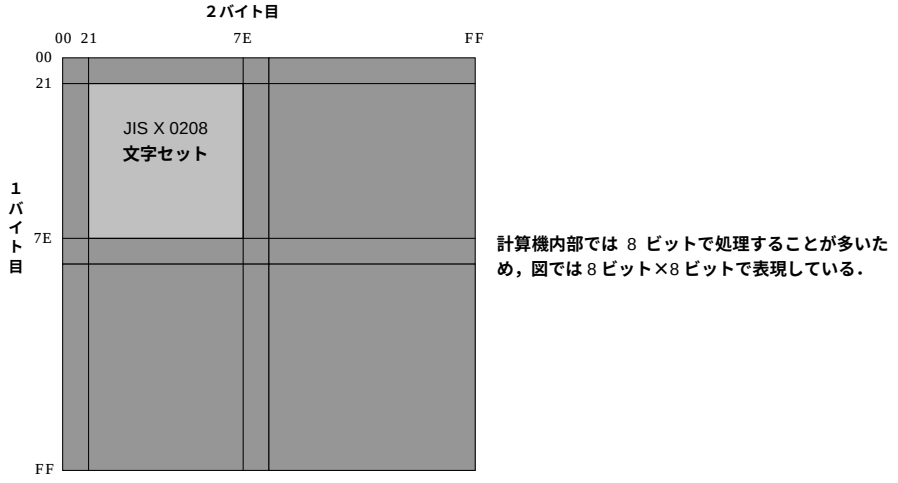

図３ JIS **コード**

また，先に出てきた番号 32h 43h は ASCII や JIS ローマ字では“ 2C ”と解釈できます．この曖昧さを解決するためにエスケープシーケンスを用います．エスケープシーケンスとは「これから先は JIS ローマ字を使います」というような指示を示す文字列です．エスケープシーケンスの例を表 2 に示します．たとえば，“ abc あいう 123 ”を JIS コードであらわすと 61h 62h 63h 1Bh 24h 42h 28h 65h 28h 66h 28h 67h 1Bh 28h 42h 31h 32h 33h となります（図４）．このうち，1B 23 42 のところが，ここから JIS X 0208 を使いますという印，1B 28 42 のところが，ここから ASCII を使いますという印になっています．なお，エスケープシーケンスがなければ，行頭は ASCII コードで始まるという決まりがあります．

文字列	**a**	**b**	**c**	**ESC**	**$**	**B**	あ		い		う		**ESC**	**(**	**B**	**1**	**2**	**3**
番号列	**61**	**62**	**63**	**1B**	**24**	**42**	**28**	**65**	**28**	**66**	**28**	**67**	**1B**	**28**	**42**	**31**	**32**	**33**

図４ JIS **コードによる表現**

ISO-2022-JP は JUNET のメンバが情報交換用の文字コードとして RFC1468 に登録したもので，表３に示す文字集合を扱うことができます．RFC とは ITIF と呼ばれる組織がまとめているインターネットに関する技術文書群です．

ISO-2022-JP を元に RFC1554 には，中国文字，補助漢字，韓国文字，ラテン文字等を利用可能にした ISO-2022-JP-2 が，RFC2237 には補助漢字を利用可能にした ISO-2022-JP-1 が登録されています．また，RFC1468 ではその使用を認められていない半角カタカナは，Microsoft Outlook などでは，ESC (I を JIS カタカナ用のエスケープシーケンスとして無理矢理使用しています．

表３ エスケープシーケンスの例

エスケープシーケンス	文字コード
ESC (B	ASCII
ESC $ B	JIS X 0208
ESC $ @	JIS C 6226-1978（未説明）
ESC (J	JIS X 0201（JIS ローマ字）

3.4 Shift_JIS コード

日本のパーソナルコンピュータで多く使われている文字コードが Shift_JIS です．Shift_JIS は 8 ビットの番号二つの組み合わせ（２バイト）で JIS X 0201（JIS ローマ字と JIS カタカナ）と JIS X 0208 の文字セットを表現します．そして，1 バイト目に JIS8 の空いている番号を使うことで，その番号が現れた場合は漢字であると判定できるようにし，エスケープシーケンスなしでローマ字などの 1 バイト文字と漢字などの 2 バイト文字を共存させています．

具体的には 1 バイト目には 81h～9Fh と E0h～EFh を使用し，2 バイト目には制御文字の範囲と FDh～FFh を避けた 40h～7Eh と 80h～FCh を使用し，47×188=8826 種を表現します．JIS コードからシフト JIS への変換は単純な変換式で行うことができます（図５）．

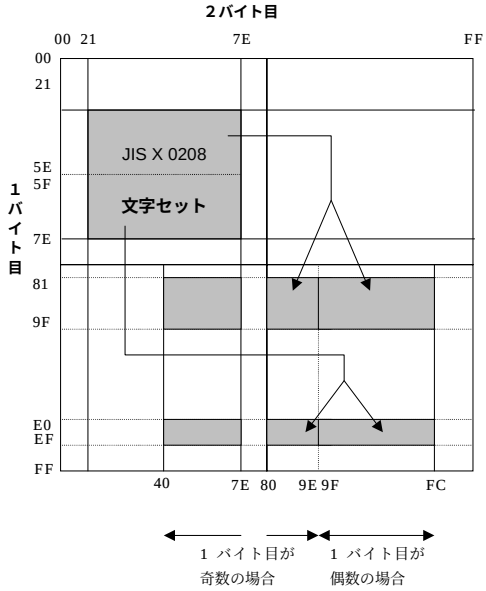

図5 JIS **コードからシフト** JIS **への変換図**

3.5 EUC コード

EUC コードは日本語対応の UNIX で広く用いられています．EUC も JIS コードと同様に JIS X 0208 の文字セットを表現します．EUC にはデータ長が固定のものと可変のものがありますが，一般的なのは可変長のものです．1 バイト目でどの文字コードを利用するかを指示することでデータ長とその解釈を変え，複数の文字コードをあらわすことができるようになっています（表4）．

具体的には，1 バイト目の 8 ビット目が 0 の場合は 1 バイト目を ASCII コードと解釈します．1 バイト目が A0h～FFh までの場合は 1 バイト目と 2 バイト目の 8 ビット目を 0 としたものを JIS コードとして解釈します．1 バイト目が 8Eh（制御コード SS2）の場合，2 バイト目を JIS カタカナと解釈します．1 バイト目が 8Fh（制御コード SS3）の場合，2 バイト目と 3 バイト目を一般には補助漢字と解釈します．

表4 EUC **コード**

1バイト目	2バイト目	3バイト目	文字コード
21h～7Eh	-	-	ASCII
A0h～FFh	A0h～FFh	-	JIS X 0208
8Eh	A0h～FFh	-	JIS カタカナ
8Fh	A0h～FFh	A0h～FFh	JIS X 0212（補助漢字）

3.6 Unicode

Unicode は ISO が制定した世界中の主な文字を扱う文字セット規格 UCS の 2 バイト方式によるもので，256×256=65536 種を表現できます．その中に日本語，中国語，韓国語で使われる漢字が含まれています．ただし，各国語の漢字すべてを入れることはできないので，形が似ているものは統合しています．Windows2000 や Java ではすでに Unicode を利用しています．

4. 改行コード

ファイルに記録された文字データを読み出したり，逆に，文字データをファイルに記録したりするとき，文字コードと共に気をつけなければならないことに改行コードがあります．改行 k コードとは，その言葉が示すとおり，行を改める場所を示すコードです．これがなぜ気をつけなければならないのかというと，OS によって改行コードが異なるからです．現在用いられている代表的な OS である，MS-Windows，UNIX，MAC の改行コードを表５に示します．どの OS で作られたファイルであるかがわからないときは，どの改行コードがきても正常に動作するようにしておく必要があります．

表５ 改行コード

OS	改行コード
Windows	0D 0A
UNIX	0A
Mac	0D

5. 課題

準備：文字コード表示

次のプログラムはキーボードから入力された文字，および，ファイルに記録された文字のコードポイントを表示するプログラムです．実際に入力して動かしてみましょう．

```
#include <stdio.h>
void main()
{
    int c;
    while( (c = getchar()) != EOF )
    {
        printf(“%02X”, c );                    /* 16 進 2 桁で表示 */
        if( c == ‘\n’ ){ printf( “\n” ); }
    }
}
-------------------------------------------------------------------------------
#include <stdio.h>
void main( int argc, char** argv )
{
    FILE* fp;
    int   c;
    if( argc > 1 )
        if( argv[1] != NULL )
            if( (fp = fopen( argv[1], "r" )) != NULL )
            {
                while( (c = fgetc( fp )) != EOF )
                {
                    printf("%02X ", c );              /* 16 進 2 桁で表示 */
                    if( c == '\n' ){ printf( "\n" ); }
                }
                fclose( fp );
            }
}
```

課題 1：大文字小文字変換

引数として渡された文字列（char 型へのポインタ）に含まれるアルファベット(A～Z, a～z)の大文字を小文字に，小文字を大文字に変換する関数

```
void change( char* str )
```

を作成しなさい．引数として与えられる文字列の文字コードは ASCII コードとします．なお，アルファベット以外の文字については変換せずにそのままにすること．また，引数として渡される文字列の終端には NULL 文字（'¥0'）が格納されていることとします．

課題 2：数字列→数値変換

引数として渡された数字列（char 型へのポインタ）を整数値に変換し int 型で返す関数

```
int atoi_func( char* str )
```

を作成しなさい．引数として渡される文字列の終端には NULL 文字（'¥0'）が格納されていることとします．また，整数を構成する文字（0～9，先頭のみ+,-）以外の文字が現れたら，数字列はその手前で終わりと判断して変換を行うこととします．たとえば，"-123a"という数字列は-123 という負の三桁の整数に変換することになります．

課題 3：JIS コード文字数カウント

JIS コード（JI-2022-JP）を使用した入力ファイルに含まれる文字数を数えて，その数を表示するプログラムを作成しなさい．空白（全角空白は除く）や改行など文字ではないものは数えないようにすること．

課題提出方法

レポートの締め切り，提出方法等は講義中に説明します．

6. おわりに

本演習では文字コードについて学びました．いろいろなプログラムを作る上で文字コードの知識はとても重要です．特に漢字を扱う場合，文字コードの特性を理解しておくことがとても重要です．この機会に文字コードについていろいろ調べてみましょう．