コンパクトな文字発生方式について*

鈴木 隆一** 池田 克夫** 清野　武**

Abstract

This paper discusses a technique for the compact generation of all symbols adopted in ISO-code. Symbols similar to those used in ordinary print are generated, requiring 32 bits per symbol, by using frames which consist of fifteen basic constructional points. These points are determined by both the symmetry of symbols and the biased distribution of characteristic points in each set of symbols such as alphanumerics, kanamoji, etc.

Symbols are generated, by selecting an appropriate frame according to shift-codes, bit-patterns, and special codes, and by connecting less than eight constructional points.

1. まえがき

文字発生用の装置は，すでに多く開発されているが，利用できる文字は発生可能なものだけであり，それ以外の文字を発生する場合には，ソフト的に行なう必要が生じる．また，そのような製置がない場合には，すべてソフト的な方法によらなければならない．

ソフト的に文字を発生する場合，その形に対応するデータを，前もって記憶しておく必要がある．とくにわが国においては，文字として英数字や記号に加えて，カナ文字までが使われることを考えると，その種類（個々の文字をいう）は非常に多くなる．そこでメモリーを有効に使うために，整数倍，できれば実際の処理が，簡単な2の巾乗の語数によって，一文字あたりのデータを格納することが望まれる．ISOコードなどでは，8ビット（バイト）を一文字の符号に割りあてていることより推察して，ほぼ4バイト（32ビット）程度のデータ量を一文字に対して用いることを目標にして考える．

ソフト的に文字を発生する方法は，現在までにも，いくつか報告されているが，文字を構成するための格子点として，常に規則的な配置のものを考え，（1）その座標を用いて各点を区別する方法，（2）規則的な格子点の数を減らし，その位置に相当する符号で，各点を区別する方法がある．前者においては，各文字の形はきれいであるが，一文字あたりに必要なデータは多く，後者においては点の配置に大きな制限があるために形は見にくくなるが，一文字あたりに必要なデータは少なくてすむ．この両者の長所をおのおの生かした処理方法について論じたのが小文であり，発生された文字の形に最も重点を置き，条件の許す範囲で，その形が通常用いられる文字に近くなることを念頭においた．対象とした文字は，ISOコードに採用されているすべての文字および記号である．以下，英字の大文字，数字およびカナ文字に対する基本的な考え方，処理法を2., 3. で述べ，その拡張として，特殊記号，小文字および拗音の処理法を4. で述べる．

2. 文字の形状による構成点の選出

2.1 文字の形状

2.1.1 文字の対称性　文字をその形状により大別すると

1) 対称な文字
   a) 上下対称　*B*, *C*, *D*, *E*, *K*, 3
   b) 左右対称　*A*, *M*, *T*, *U*, *V*, *W*, *Y*
      ハ, ホ, ニ
   c) 上下，左右対称　*H*, *I*, *X*, 0, 8
2) 非対称な文字
   英数字18種，カナ文字43種

となる．対象とした文字の処理のためには，上に述べた範囲の対称性を考えれば充分である．

文字の大部分は非対称の類に属し，対称な文字として取り扱えるものはあまり多くない．また，カナ文字には対称なものとして，左右対称のものだけが認めら

* A Techniqure for the Compact Generation of Sybols, by Ryuichi SUZUKI, Katsuo IKEDA, Takeshi KIYONO (Faculty of Engineering, University of KYOTO)

** 京都大学・工学部

れる．

2.1.2 文字を構成する特徴点の分布

文字は，その筆順が示すように

1) 見える線分

2) 隠れた線分

の二つのモードの線分が，特定の順序で結ばれたものと考えることができる．この場合，見える線分の連続することがきわめて多い．

文字を構成する線分の数は，線分のとらえ方，曲線部の近似の程度により異なる．一般に，対称な文字およびカナ文字は，非対称な文字よりも多くの線分で構成されている．

つぎに，文字を線分で近似したときの各線分の端点を特徴点と呼ぶことにし，文字はこの特徴点が適当な順序およびモードで結ばれて構成されており，その形のみが意味を持つものと考える．このような特徴点の分布を考える場合，その分布は文字の形状および分類（英数，カナ，記号のような区分をいう）により異なると考えられるので

1) 非対称　英数字

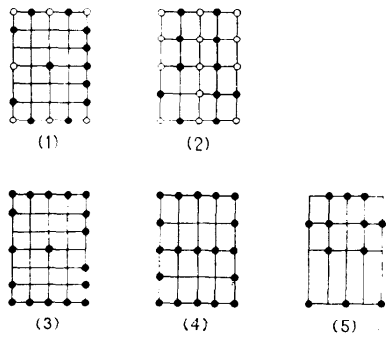
2) 非対称　カナ文字

3) 上下対称　英数字

4) 左右対称　英数字

5) 左右対称　カナ文字

の類別に従って調べてみると，**Fig. 1** に示される形となる．



(1) asymmetric alphanumerics (2) asymmetric kanamoji
(3) alphanumerics with horizontal axis of symmetry
(4) alphanumerics with vertical axis of symmetry
(5) kanamoji with vertical axis of symmetry

Fig. 1 The distribution of the characteristic points

1) は非対称な英数字に対する特徴点の分布で，全体的には均一にならず，○で囲んだ点を中心として，その周辺近傍に多く分布し，●で示された点を中心としても分布することが認められる．

2) は非対称なカナ文字に対する特徴点の分布で，全体的な疎密はあるが，均一に特徴点が分布する傾向がある．

3) は上下対称な英数字に対する特徴点の分布で，1)（非対称な英数字の特徴点）の一部を取り出した形になっている．

4) は左右対称な英数字に対する特徴点の分布で，2)（非対称なカナ文字の特徴点）の一部を取り出した形になっている．

5) は左右対称なカナ文字に対する特徴点の分布で2)（非対称なカナ文字の特徴点）の一部を取り出した形になっている．

2.2 文字の表現

2.2.1 文字の量子化　前章で得られた結果より，文字を構成する特徴点の分布には，全体的にかなりの程度の集中化が認められる．このことは逆に適当に選び出されたいくつかの点を，2種のモードの線分を用いて連結することにより，各文字が構成できることを示している．このような観点より文字をとらえた場合，量子化された特徴点の集合から，どれだけの点を基本構成点として採用し，それらのうちのいくつを構成点として選べば，すべての文字を表現できるかが問題となる．文字の形は構成点が多いほどきれいであるが，必要なコアの語数は増大するので，適当な妥協点を見い出す必要がある．以下，文字の形と実際に必要なデータ，および処理との両面より考察する．

2.2.2 データ形式　基本構成点（適当に選ばれた特徴点の組）の中から，制限された個数の構成点を選び，見える線分または隠れた線分（モード指定）で結び文字を構成する．その方法として

1) 1つの点を表わすデータにおいて，1ビットをモード指定用に，残りのビットで各点の X，Y 座標を与える形式．

2) 1つの点を表わすデータにおいて，1ビットをモード指定用に，残りのビットで各点の位置を示す符号を与える形式．

3) 点の位置として割りあてられていない符号をモード指定のために割りあて，符号の並べ方に一定の規則を与える形式．

の3種の形式が考えられる．各形式の長短は，必要とされる文字の種類およびデータの性質によって決まる．いま，対象としている文字は

ⅰ) 種類が多い.
ⅱ) 特徴点の分布に偏りがある.
ⅲ) 線分のモードに偏りがある (見える線分から見える線分に続いている場合が多い.).

などの特徴を持っている. このことより, 3) に示した形式が最も適していると考えられる.

1) の形式においては, 各点の座標を示す部分の情報密度は低く, データを格納するために必要なビット数は多いが, そのデータが直接使用できる点に利点があり, 要求される文字の種類が少ない場合や, 必要な点がランダムに存在している場合には適している. しかし, 対象として考えている文字の種類が多く, 文字の構成に必要な点にある程度の偏りがある場合には, 1) の形式は不適当である. 2) および 3) の形式では, 記憶される点の位置に関するデータは, 最高の情報密度を持つようにできるが, 与えられるデータは, 関接的なものであり, 実際に必要なデータ算出のためのデータと処理は多くなる. この増加の割合と, この形式により減らすことのできるデータの量を比較したとき, 1) と 2) および 3) の形式のいずれが適しているかが決定される. さらに, モード指定用に1ビットを割りあてることについて検討してみる. 実際に対象としている文字においては, その筆順を適当に選べば, 二つのモードは必ずしも同じ割合には現われず, 見える線分に相当するモードがはるかに多い. このことは1ビットをモード指定に割りあてることが, 必ずしも最大の情報密度を与えていないことを示している. そこで 3) の形式が考えられ, モード指定も点を示す符号と同様に1符号を割りあて, その符号のつぎのデータは, 隠れた線分であると決めておく. 英数字およびカナ文字など多くの文字を扱う場合には, 3) の形式を採用するのが, 最も適していると考えられる.

2.2.3 文字の形状の表現 対称な文字は一般に表現のために, 非対称な文字よりも多くの構成点を必要とするが, 対称という概念を用いると, 1文字に対して必要なデータは半分ですみ, きわめて都合がよい. 文字の分類および形状によって, 特徴点の分布が異なることは前節で述べたが, その結果を用いて, つぎのように文字を類別し, それぞれについて適当な特徴点の組 (以下枠という) を与えたほうが, よりコンパクトな表現とすることができる.

1) 非対称な英数字.
2) 非対称なカナ文字.
3) 上下対称な英数字.
4) 左右対称な英数字およびカナ文字.

前節の考え方を用いれば, 個数の少ない対称な文字に, 対称性を示す特定の符号または符号の並びを割りあて, 無指定ならば非対称であることにするのが適している. 対称性は文字をかく前に, その形状を判別する必要があるので, 最初に現われる符号をその形状に対する符号と考える. この符号が対称性 (上下, 左右対称) を示すもの以外であれば, 非対称であり, それが点の位置を示すとみなす. 本文においては, 2種の対称性を考えているが, 最初にくる点の位置を制限することにより, 点の位置を示すために与えられた符号を, 対称性を示す符号として割りあてることができる.

つぎに, 非対称な文字に含まれる英数字とカナ文字の区別は, シフト記号 (ISO コードの SI/SO 符号) によるフレームの切換えの概念を用いて, 容易に実現される.

2.2.4 文字を構成する要素間の関係 本章で述べてきた結果を総合すれば, つぎのような結論を得る. 文字用データを記憶するコア・メモリーが1語 B ビットよりなり, m 語を使用するものとすれば, 文字を表現するために選び出された基本構成点を P 個, 1文字を表現するために使用できる最大の構成点の個数を N 個として, モード指定用に1符号を用いることを考慮すれば

$$N \leqq \frac{m \cdot B}{\lceil \log_2(p+1) \rceil} = \frac{\text{〔全ビット数〕}}{\left[\begin{array}{c}\text{一つの点の位置を示す}\\ \text{ために必要なビット数}\end{array}\right]}$$

$\lceil n \rceil$ は n より小でない最大の正整数とする.

なる関係があり, 各等式が成り立つときに, 最も情報量は大きい.

3. コンパクトな文字の発生法

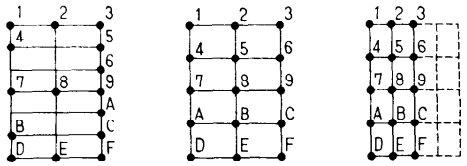
3.1 文字を構成する各要素の値

2.2.4 で述べた要素の値を求めるために, 前に求めた特徴点の中の一部をとり, 基本構成点の数 (p) と構成点の数 (N) を, コアの語数 (m) を変化させて文字をかいてみて求めると, $m=2$, $p=15$, $N=8$ がその妥協点として得られた. すなわち, 1文字あたり2語 (32ビット) で, 基本構成点を適当にとれば, 形をあまりくずさずに各文字が発生できる. この場合, 本質的に画数の多いカナ文字の一部に, やや形のくずれるもの (サ, チ, ネなど) があるが, その固有のイメージはそこなわれていないと考えてもよい. 以下, 本章においては, 基本構成点の選択法, 具体的な処理

法について述べる．

3.2 基本構成点の選出および符号化

2. で求めた各形状に対する特徴点の分布のうち，同一の類において，なるべく多くの文字が日常用いられる形に近くなるように，しかも，等間隔の格子の上にあるという条件のもとに選んだ15点を基本構成点と呼ぶ．特徴点の分布 (Fig. 1) から，**Fig. 2** に示したよ



(a) alphanumerics with horizontal axis of symmetry and asymmetric alphanumerics
(b) asymmetric kanamoji
(c) alphanumerics and kanamoji with vertical axis of symmetry

Fig. 2 Frames consisting of fifteen basic constructional points

うな基本構成点を選び，各類に対する枠とする．(a) は非対称および上下対称の英数字用の枠，(b) は非対称なカナ文字の枠，(c) は左右対称な文字用の枠である．各枠における基本構成点には，図に示されるような番号（1桁の16進数 1～*F*）をつけ点の位置を示す符号とする．さらに符号“0”をモード指定符号とする．

3.3 枠の選択

枠の選択は，各文字のデータの先頭の符号が“0”であるときは上下対称，“4”であるときは左右対称であることを示す．これ以外の符号が先頭にあるときは，その文字は非対称であると判断し，その符号が構成点の位置を示していると解釈する．したがって，左右対称以外の文字は，第1番目の符号が“4”以外でなければならない．これは筆順を適当に変更することにより，実用的には全くさしつかえがない．

非対称な英数字用の枠と非対称なカナ文字用の枠との切換えは，シフト記号を用いることにより，容易に行なうことができる．ISO コードを用いれば，シフト・アウト符号 (16進法の *E*) でカナ文字用の枠，シフト・イン符号 (16進法の *F*) で英数字用の枠に切換えることができる．

3.4 文字の発生

文字を構成する線分のモード指定は，**3.2** に述べたように，符号“0”により行なう．“0”のつぎにくる符号に対応する点に至る線分は隠れた線分であるとする．“0”以**外**の符号の連続は，すべて見える線分で連結されると解釈する．

符号で指定された点の座標は，各枠の基本構成点の座標を記入した表を参照して求められる．この表は，各点に対して1語16ビットを割りあて，前半の8ビットに *X* 座標，後半の8ビットに *Y* 座標が記入される．文字の拡大は，この表に与えられたデータに適当な係数を掛けることにより，任意に行なうことができる．

対称な文字の対称部（基本部分とそれに対称な部分とに分け，後者の部分をいう）の座標に対する処理は，与えられた点の符号に一定の数（基本構成点として15点をとれば16）を加えて，表より求めることができる．

1文字に対するデータは，非対称な文字に対しては8点を処理した場合，もしくはモード指定用の符号“0”が連続した場合に終了する．対称な文字は，データが終了したとき対称部の処理に移る．対称な文字に対しては約2倍の処理時間が必要となるので，できるだけ非対称な文字と同様に取り扱ったほうが合理的である．最終的には，対称な文字として処理する必要のあるのは，つぎの文字である．

1) 上下対称な文字 *B*, 3
2) 左右対称な文字 *A*, *I*, *W*, 8
ハ，ホ，ニ

4. 記号，小文字，拗音に対する拡張

4.1 記号に対する処理

記号は英数字およびカナ文字の両方に含まれる．まず，これらの記号が前章で求めた4つの枠の適当なもので構成できないかどうかを考える．英字用の記号については，非対称英数字および両対称形の枠，カナ文字用の記号は非対称カナ文字，および左右対称の枠でおのおの表現できるかどうかを検討してみる．ⓐを除いて，ほぼ満足のいく形に表現可能で，ⓐについても□をもってその直感的イメージを保ったまま表現できるものと思われる．

4.2 小文字に対する処理

小文字に対しては，前述の4種類の枠を適用することはできない．ISO コードにおいて，パリティ・ビットを除いた上3ビットのパターンが，小文字に対して同一であることに着目し，これによって小文字用の枠

$B_4 \sim B_1$ \ $B_8 \sim B_5$	0010	0011	0100	0101	0110	0111	0010	0011	0100	0101
0000	SP 0-0	0 5-2-4-B E-C-5-0	@ E-D-1-3 F-E-8-5	P D-1-2-5 6-8-7-0		P 4-D-2-3 6-8-A-9	SP 0-0	— 7-9-0-0	タ 8-7-2-3 D-0-0	≡ 1-9-0-C 4-0-7-F
0001	▽ 4-3-2-C 0-F-0-0	1 F-D-E-2 4-0-0	A 4-D-3-0 8-9-0-0	Q C-5-2-4 B-E-C-8	a F-3-6-2 4-9-D-A	Q 4-F-4-3 5-7-A-B		ア 1-3-8-0 5-B-D-0	チ 3-7-9-0 5-B-D-0	4 2-D-C-0 9-F-0-0
0010	｜ 2-8-0-0	2 C-F-D-B 6-5-2-4	B 0-1-2-5 6-8-7-1	R D-1-2-5 6-8-7-F	b 0-1-D-E C-9-6-5	r 1-B-0-7 2-3-0-0	「 3-2-E-0 0	イ 3-7-0-5 E-0-0	ツ 1-4-0-2 5-0-3-D	メ 3-D-0-4 C-0-0
0011	7 2-B-C-0 E-5-4-0	3 0-7-8-6 5-2-4-0	C 5-2-4-B E-C-0-0	S 5-2-4-7 9-C-E-B	c 6-2-4-9 D-A-0-0	s 6-2-4-7 8-A-D-9	」 2-E-D-0 0	ウ 2-5-0-7 4-6-9-E	テ 1-3-0-4 6-5-8-D	キ 7-9-0-1 3-2-E-F
0100	4 B-C-4-5 0-2-E-0	4 C-B-2-E 0-0	D 1-D-E-A 6-2-1-0	T 1-3-0-2 E-0-0	d 0-3-F-E B-8-6-7	t 0-5-7-0 4-E-F-0	、 B-E-0-0	エ 8-3-1-0 2-E-D-F	ト 1-D-0-7 9-0-0	ヤ 8-6-4-0 2-E-0-0
0101	5 D-3-8-2 4-C-E-8	5 3-1-7-8 A-C-E-B	E 3-1-D-F 0-9-7-0	U 1-B-E-C 3-0-0	e 7-8-6-2 4-9-D-A	u 1-9-D-A 3-F-0-0	・ E-0-0	オ 6 4-0-2 E-0-D-5	ナ 6-4-0-2 8-D-0-0	ユ D-F-0-0
0110	6 F-1-2-D C-0-0	6 5-2-4-B E-C-9-7	F 3-1-D-0 7-9-0-0	V 1-E-3-0 0	f 0-3-2-E 0-5-7-0	v 1 D 3-0 0	ヲ 1-3-9-D 0-7-9-0	カ E-6-4-0 2-D-0-0	ニ 4-5-6-0 A-C-0-0	ヨ 1-3-F-D 0-7-9-0
0111	7 E-3-1-4 0-0	G 5-2-4-B E-C-9-8	W 4-1-E-3 0-0	g 4-D-F-4 3-5-9-B	w 1-C-2-E 3-0-0	ア add '7'	キ 6-7-0-A 9-0-2-E	ヌ 1-3-D-0 4-C-0-0	ラ 1-3-0-4 6-D-0-0
1000	「 3-2-E-F 0-0	8 4-3-4-8 9-8-A-F	H 1-D-0-7 9-0-3-F	X 1-F-0-3 D-0-0	h 6-1-D-0 5-7-F-0	x 1-F-0-3 B-0-0	イ add 'イ'	ク 7-2-3-D 0-0	ネ 2-4-6-A 0-E-8-F	リ 1-7-0-3 9-E-0-0
1001	」 1-2-E-D 0-0	9 B-E-C-5 2-4-7-9	I 4-2-3-F E-0-0	Y 1-8-3-0 8-E-0-0	i 0-4-0-6 E-0-0	y 4 2-A-0 4-D-0-0	ウ add 'ウ'	ケ 2-4-6-0 5-B-D-0	ノ 3-6-D-0 0	ル 1-D-0-2 E-9-0-0
1010	＊ 2-E-0-5 B-0-4-C	： 4-6-0-C 0-0	J 2-3-C-E B-0-0	Z C-F-D-3 1-4-0-0	j 4-1-0-3 C-D-0-0	z A-F-B-3 1-4 0-0	エ add 'エ'	コ 4-6-F-D 0-0	ハ 4-2-D-0 0	レ 1-D-9-0 0
1011	＋ 2-E-0-7 9-0-0	； 4-6-0-C F-0-0	K 1-D-0-3 7-F-0-0	[0-1-D-0 7-A-F-0	k 0-1-D-0 7-A-F-0	オ add 'オ'	サ D-9-3-6 4-0-2-8	ヒ 1-D-F-0 9-7-0-0	ロ 1-D-F-3 1-0-0
1100	ー 8-E-0-0	＜ 5-7-C-0 0	L 1-D-F-0 0	¥ 0-2-E-0 0	l 0-2-E-0 0	ヤ add 'ヤ'	シ 1-5-0-8 4-0-D-9	フ 1-3-D-0 0	ワ 7-1-3-9 D-0-0
1101	― 7-9-0-0	＝ 4-A-C-0 6-4-0-0	M D-1-E-3 F-0-0	] B-1-4-2 D-5-3-F	m B-1-4-2 D-5-3-F	ユ add 'ユ'	ス 1-3-D-0 8-F-0-0	ヘ 7-5-C-0 0	ン 1-5-0-6 F-0-0
1110	・ E-0-0	＞ B-9-4-0 0	N D-1-F-3 0-0	↑ B-1-4-3 F-0-0	n 6-2-4-9 D-A-6-0	ヨ add 'ヨ'	セ 9-6-4-0 2-E-F-0	ホ 4-F-3-6 4-0-8-D		
1111	／ D-3-0-0	？ E-8-6-5 2-4-0-0	O 5-2-4-B E-C-5-8	← D-F-0-0	o 6-2-4-9 D-A-6-0	ッ add 'ッ'	ソ 1-5-0-3 D-0-0	マ 1-3-6-B 0-7-F-0		

add {アメ} denotes the address of アメ

Table. 1 Generated patterns and data based on ISO-code

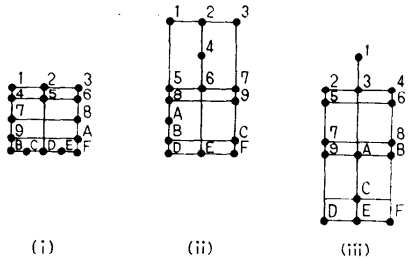


Fig. 3 Frames for lower case letters

を選択することにすれば，新しい枠が設定可能となる．枠は各文字が必要とする領域によって，つぎの3とおり必要である（**Fig. 3** 参照）

ⅰ） a c e m n o r s u v w x z

ⅱ） b d f h i k l t

ⅲ） g j p q

各枠の指定の方法は，大文字に対する対称性の場合と同様にして，データの第1番目の符号が“0”のときⅱ）に対する枠を，“4”のときⅲ）に対する枠を，他の場合にはⅰ）に対する枠を選択するようにする．

4.3 拗音に対する処理

拗音は普通のカナ文字と相似であり，その大きさおよび位置が異なる．したがって

1) ビームの位置を調整し，カナ文字用のデータから算出した座標を縮少する方法，

2) 普通のカナ文字用のデータのみを使用し，拗音用の枠を用意する方法，

3) 拗音用の枠およびデータを用意する方法，

といった方法が考えられるが，実際の処理を考慮した場合，2）の方法が適していると考えられる．

拗音用の枠は，前節と同様に，パリティ・ビットを除いた上位3ビットおよび下位4ビットのビット・パターンで選択する．

5. ま と め

一定の制限の範囲で，最もよい形の文字を得るために，1) 文字の形状を調べ，2) 文字を特徴点の分布および対称性などにより分類し，3) おのおのに則して限られた数の基本構成点を選び，4) シフト記号，ビットパターン，特定の符号により枠を切換える文字発生方式について述べた．本方式によると，1文字あたりのデータを32ビットに格納することができ，ISO コードに含まれるすべての文字が，その形をあまりくずさずに発生可能であるので，ISO コードに対する文字発生の1つの標準的な方式として適当であると考えられる．

コンパクトな形の文字発生ということで，文字の形に制限を加えざるを得ないのであるが，このようにして発生された文字が，それを見，読む人にとって，どのような影響を与えるか，とくに，拡大された場合，長時間そのような文字に接する場合など，人の心理をも含めたときの情報伝達に関して，各種の問題があるものと思われる．本方式はこのような問題に対しても，かなり有効な手段を提供するものと思われる．

本研究は，文部省科学研究費補助金の交付を受けている研究の一環として行なったものである．

参 考 文 献

1] 戸川隼人；線画で文字を表示する場合の元データを1字当り1語で表現する方法，情報処理学会，昭和44年度第10回大会講演予稿集．

2] D. C. HITT, G. H. OTTAWAY and R. W. SHIRK; The mini-computer-A new approach to computer design. FJCC, Vol. 33, 1968, pp. 655.

（昭和 45 年 6 月 12 日受付）