入学年度	学部	学 科		組		番 号		検	フリガナ
									氏 名

QR コードの仕組みを詳しく知るために，例として学籍番号を実際に QR コードにしてみよう．

- QR コードの型

中大の学籍番号は 11 文字なので，一番小さい 1 型を選ぶ．この文字数なら誤り訂正レベルが高めの「レベル Q」を選択しても 1 型に収まるので，1-Q 型を選ぶことにする．

- データの bit 列化

QR コードでは，数字，英数字，8 bit バイト，漢字などのモードが用いられる．学籍番号は英数字なので「英数字モード」を用いることにする．これを指示するため，まず最初の 4bit は 0010 とする．

つぎに 中大の学籍番号は 11 文字なので，これを 2 進法で表示すると 1011 となる．英数字モードの場合の最大格納文字数の関係から，これを 9bit で表し 000001011 とする．

そして，いよいよ実際のデータを bit 列になおす．英数字モードではまず下表の通りに各文字を数字化する．なぜ 45 文字が使用可能かというと，$45^2 = 2025 \fallingdotseq 2048 = 2^{11}$ なので，2 文字の組を 11bit で表すことができ，効率よく符号化できるからである．そこで，データを 2 文字ずつに区切り，1 つ目の文字の下の表の値を 45 倍したものと 2 つ目の文字の表の値を足す．（2 文字の並びは「45 進法」で表されていると考え，これを 10 進法に直すことに相当する．）

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I	J
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		$	%	*
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
+	-	.	/	:															
40	41	42	43	44															

たとえば，“17” は $1 \times 45 + 7 = 52$，“B1” は $11 \times 45 + 1 = 496$ となる．なお文字数が奇数の場合は最後に残った 1 文字は対応する値をそのままの値とする．さらに，こうして計算された数を 11bit の 2 進数で表す．たとえば，“52” は 2 進法で 110100 だが，これを 11bit にするために最初に 0 をいくつか加え，00000110100 とする．以下，これをこれを続け，文字数が奇数の場合の最後に残った 1 文字は対応する値を 2 進法で表し，6 bit で表記する．

学生証番号	1	7	B								
「10 進法化」	52										
11bit 化											

すべてを bit 化したら，最後に終端パターンとして 0000 を付加する．

$$\underbrace{0010}_{\text{英数字モード}}\ \underbrace{000001011}_{\text{文字数 11}}\,\underbrace{00000110100}_{\text{“17”}}\ \underbrace{\cdots}_{\text{“B・”}}\quad \cdots \quad \underbrace{0000}_{\text{終端パターン}}$$

こうして得られたデータを 8bit ごと（1byte ごと）に区切り直す．最後のビット列が 8bit 未満の場合は 0 で埋める．また，1-Q 型では RS(26, 13, 6) 符号を用いるので，得られた byte 数が情報 byte 数である 13 に満たない場合は ”11101100” および ”00010001” という「埋め草パターン」を交互に付加しする．

英数字モード				文字数 11									“17”										
0	0	1	0	0	0	0	0	0	1	0	1	1											
“B・”																							
		終端パターン				0 fill		埋め草パターン 1								埋め草パターン 2							
		0	0	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	1	0	0	0	1
埋め草パターン 1																							
1	1	1	0	1	1	0	0																

このようにして，13byte からなる情報語を得る．

	8bit データ							
1.	0	0	1	0	0	0	0	0
2.	0	1	0	1	1	0	0	0
3.								
4.								
5.								
6.								
7.								
8.								
9.								
10.								
11.								
12.								
13.	1	1	1	0	1	1	0	0

- 誤り訂正符号化

1-Q 型の QR コードでは RS(26, 13, 6) 符号と呼ばれる符号を用いる．この符号は $GF(2^8) = GF(256)$ を係数とする 25 次多項式を符号語とする符号である．$GF(2^8)$ は $GF(2) = \mathbf{F}_2$ に $\gamma^8+\gamma^4+\gamma^3+\gamma^2+1=0$ をみたす γ を付け加えた体である．

表で得られたデータの各 byte を γ の 7 次以下の多項式とみなす．すなわち, “00100000” を γ^5, “01011000” を $\gamma^6+\gamma^4+\gamma^3$ などとする．そして，13byte の情報語を，係数が $GF(2^8)$ の要素である x の 13 次の多項式とみなす．すなわち，上の情報語は

$$q(x) = \gamma^5 x^{13} + (\gamma^6+\gamma^4+\gamma^3)x^{12} + \cdots + (\gamma^7+\gamma^6+\gamma^5+\gamma^3+\gamma^2)$$

という情報多項式で表せる．そして，生成多項式 $g(x)$ を

$$g(x) = (x+1)(x+\gamma)(x+\gamma^2)(x+\gamma^3)\times\cdots\times(x+\gamma^{12})$$

として，送信多項式 $u(x)$ を $g(x)$ 用いて次のように

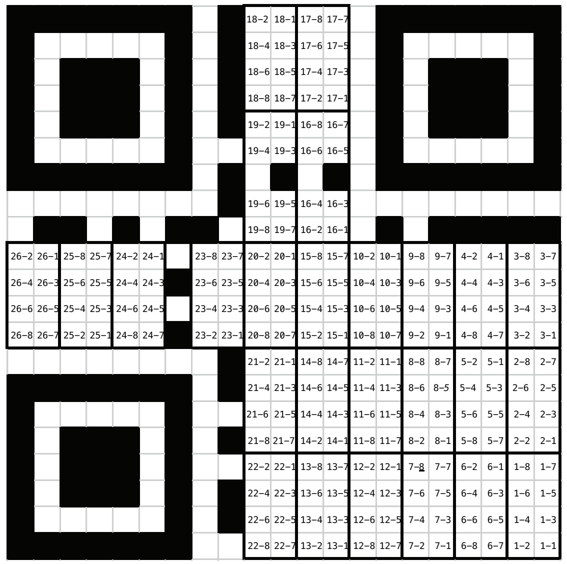
$$u(x) = q(x)x^{13} + \big(q(x)x^{13} \text{ を } g(x) \text{ で割った余り}\big)$$

$u(x)$ の計算は Mathematica のファイルを用いて行い，それを下に写す．

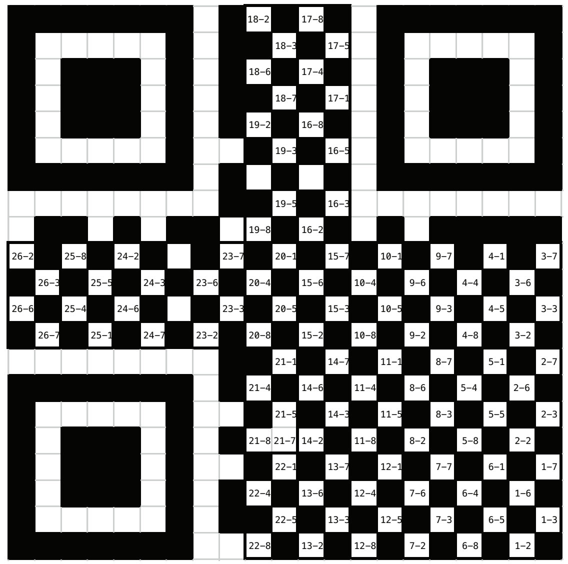
1.	0	0	1	0	0	0	0	0
2.	0	1	0	1	1	0	0	0
3.								
4.								
5.								
6.								
7.								
8.								
9.								
10.								
11.								
12.								
13.	1	1	1	0	1	1	0	0

14.								
15.								
16.								
17.								
18.								
19.								
20.								
21.								
22.								
23.								
24.								
25.								
26.								

この結果をもとに右のページの左上の図のマス目を黒く塗っていく．

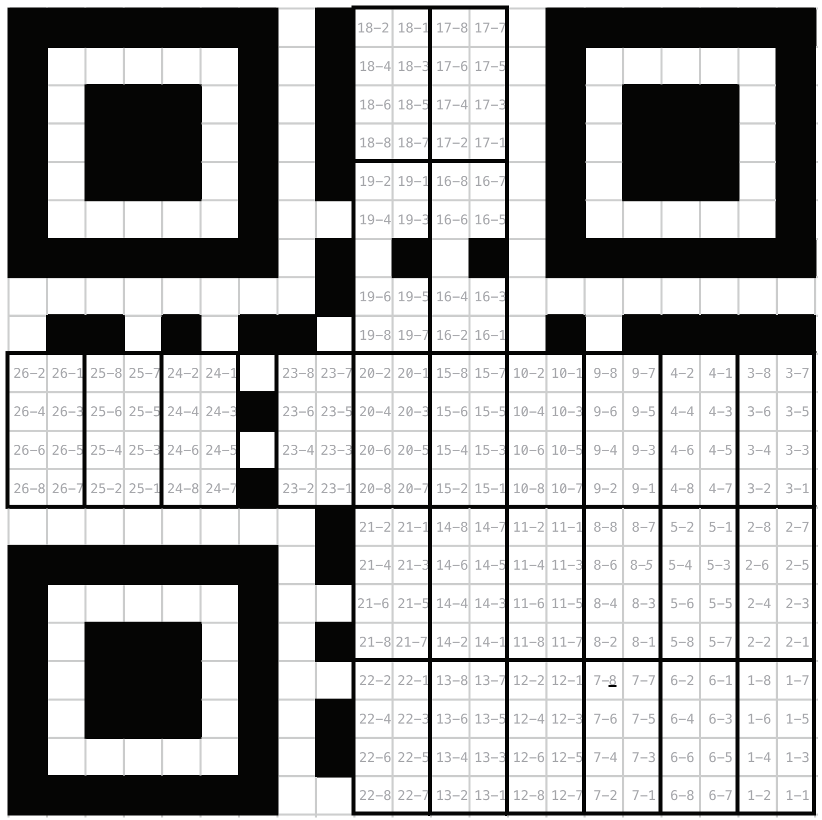


マスク前



マスク

左右の図を見比べ，「白」+「白」=「黒」+「黒」=「白」,「白」+「黒」=「黒」+「白」=「黒」として下の図を塗っていく．



担当：鍬田 政人 (kuwata@tamacc.chuo-u.ac.jp)