MATH 5: HANDOUT 11
BINARY NUMBERS. n-ARY NUMBERS.

Today we talked more about binary numbers. We discussed arithmetic operations with binary numbers.

We also touched on other bases. For example, in base 3, we only use digits 0, 1, 2, and they correspond to powers of 3:

$$21021_3 = 2 \cdot 81 + 1 \cdot 27 + 0 \cdot 9 + 2 \cdot 3 + 1 \cdot 1 = 196_{10}$$

If the base is larger than 10, then in addition to digits $0 \dots 9$ we use letters A, B, etc. For example, in base 16, we use digits $1, \dots, 9$ and letters $A = 10, B = 11, C = 12, D = 13, E = 14, F = 15$. The digits correspond to powers of 16:

$$D4B_{16} = D \cdot 256 + 4 \cdot 16 + B \cdot 1 = 13 \cdot 256 + 4 \cdot 16 + 11 \cdot 1 = 3403_{10}$$

The topic we didn't discuss (but also an interesting topic) is how letters and other symbols are written as sequences of 0 and 1 (bits) in computers; since there 2^n such sequences of length n, and there are 26 letters in English alphabet, we need at least 5 bits ($2^5 = 32$) for each English letter. If we want to have lower- and upper-case letters, punctuation, numbers, accented letters such as é, we need more; in real life, people use 8 bits per symbol (called "byte").

The correspondence between actual letters and their codes, i.e. sequences of 0 and 1, is called *encoding*. The most common encoding (Latin 1, aka ISO 8859-1) is shown in the attached table. In this table, rows correspond to the last 4 bits, and columns, to the first four bits. For example, lower case letter "a" has code 01100001.

				b8	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
				b7	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	
				b6	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
				b5	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
b4	b3	b2	b1		00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
0	0	0	0	00			SP	0	@	P	`	p			NBSP	°	À	Ð	à	ð	0
0	0	0	1	01			!	1	A	Q	a	q			¡	±	Á	Ñ	á	ñ	1
0	0	1	0	02			"	2	B	R	b	r			¢	²	Â	Ò	â	ò	2
0	0	1	1	03			#	3	C	S	c	s			£	³	Ã	Ó	ã	ó	3
0	1	0	0	04			$	4	D	T	d	t			¤	´	Ä	Ô	ä	ô	4
0	1	0	1	05			%	5	E	U	e	u			¥	µ	Å	Õ	å	õ	5
0	1	1	0	06			&	6	F	V	f	v			¦	¶	Æ	Ö	æ	ö	6
0	1	1	1	07			'	7	G	W	g	w			§	·	Ç	×	ç	÷	7
1	0	0	0	08			(	8	H	X	h	x			¨	¸	È	Ø	è	ø	8
1	0	0	1	09			)	9	I	Y	i	y			©	¹	É	Ù	é	ù	9
1	0	1	0	10			*	:	J	Z	j	z			ª	º	Ê	Ú	ê	ú	A
1	0	1	1	11			+	;	K	[	k	{			«	»	Ë	Û	ë	û	B
1	1	0	0	12			,	<	L	\	l	|			¬	¼	Ì	Ü	ì	ü	C
1	1	0	1	13			-	=	M	]	m	}			SHY	½	Í	Ý	í	ý	D
1	1	1	0	14			.	>	N	^	n	~			®	¾	Î	Þ	î	þ	E
1	1	1	1	15			/	?	O	_	o				¯	¿	Ï	ß	ï	ÿ	F
					0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	hex

Homework

1. What is the largest number that can be written as a 5-digit binary number? (Hint: what is the smallest 6-digit binary number?)

2. Is it possible to encode every letter of English alphabet by a 4-digit binary number? You can choose any way you like — for example encoding A as 0000, B as 0001, or by any other method.

Would it be possible if we used 5-digit binary numbers?

3. A car has traveled 125 miles during some period. During the same period, another car, which is faster by 10 mph, has traveled 150 miles. What is the speed of the faster car?

4. Do the following arithmetic operations with binary numbers. Try doing them without converting the numbers to decimal form.

(a) $110101_b + 111011_b$ (b) $10101_b \times 1011_b$ (c) $(10101_b + 1101_b) \times 10110_b$

5. Fish head weighs as much as the tail and half of the body together. The body weighs as much head and tail together, and the tail weighs 1 kg. How heavy is the fish?

6. In order to allow computers to deal with different languages, computer scientists have developed so-called Unicode, a standard list of symbols covering virtually all human languages, from Armenian to Vietnamese. In particular, it includes Latin letters, Cyrillic letters, Chinese characters (hanzi), Emoji, and more.

The latest revision of Unicode contains about 96,000 symbols. If we want to represent each of them by a sequence of 0 and 1, would it be enough to use 16 bits (0s and 1s) for each symbol? If not, what is the smallest number of bits per symbol one would need?