

6.004 Fall 2019 Tutorial Problems

L01 – Binary Encoding and Arithmetic

Note: A small subset of essential problems are marked with a red star (★). We especially encourage you to try these out before recitation.

Problem 1. Encoding positive integers

1. What is the 5-bit binary representation of the decimal number 21? ★
2. What is the hexadecimal representation for decimal 219 encoded as an 8-bit binary number?
3. What is the hexadecimal representation for decimal 51 encoded as a 6-bit binary number?
4. The hexadecimal representation for an 8-bit unsigned binary number is 0x9E. What is its decimal representation? ★
5. What is the range of integers that can be represented with a single unsigned 8-bit quantity?
6. Since the start of official pitching statistics in 1988, the highest number of pitches in a single game has been 172. Assuming that remains the upper bound on pitch count, how many bits would we need to record the pitch count for each game as an unsigned binary number?
7. Compute the sum of these two 4-bit unsigned binary numbers. Express the result in hexadecimal. ★

$$\begin{array}{r} 1101 \\ +0110 \\ \hline \end{array}$$

Problem 2. Two's complement representation

1. What is the 6-bit two's complement representation of the decimal number -21?
2. What is the hexadecimal representation for decimal -51 encoded as an 8-bit two's complement number?
3. The hexadecimal representation for an 8-bit two's complement number is 0xD6. What is its decimal representation?
4. Using a 5-bit two's complement representation, what is the range of integers that can be represented with a single 5-bit quantity?
5. Can the value of the sum of two 2's complement numbers $0xB3 + 0x47$ be represented using an 8-bit 2's complement representation? If so, what is the sum in hex? If not, write NO. ★
6. Can the value of the sum of two 2's complement numbers $0xB3 + 0xB1$ be represented using an 8-bit 2's complement representation? If so, what is the sum in hex? If not, write NO.

7. Please compute the value of the expression $0xBB - 8$ using 8-bit two's complement arithmetic and give the result in decimal (base 10).
8. Consider the following subtraction problem where the operands are 5-bit two's complement numbers. Compute the result and give the answer as a decimal (base 10) number. ★

$$\begin{array}{r} 10101 \\ - \underline{00011} \end{array}$$

Problem 3. Multiples of 4

1. Given an unsigned n -bit binary integer $v = b_{n-1} \dots b_1 b_0$, prove that v is a multiple of 4 if and only if $b_0 = 0$ and $b_1 = 0$.
2. Does the same relation hold for two's complement encoding?

Problem 4. Encoding text

There are multiple standards to encode characters and strings using binary values. ASCII is a classic standard to encode English alphabet characters (modern formats like UTF support other alphabets, but are typically based on ASCII). ASCII encodes each character using an 8-bit (1-byte) value. The table below shows ASCII's mapping of characters to values.

ASCII (1977/1986), adapted from <https://en.wikipedia.org/wiki/ASCII>

	<u>0</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>
<u>0</u>	NUL 0x00	SOH 0x01	STX 0x02	ETX 0x03	EOT 0x04	ENQ 0x05	ACK 0x06	BEL 0x07	BS 0x08	HT 0x09	LF 0x0A	VT 0x0B	FF 0x0C	CR 0x0D	SO 0x0E	SI 0x0F
<u>1</u>	DLE 0x10	DC1 0x11	DC2 0x12	DC3 0x13	DC4 0x14	NAK 0x15	SYN 0x16	ETB 0x17	CAN 0x18	EM 0x19	SUB 0x1A	ESC 0x1B	FS 0x1C	GS 0x1D	RS 0x1E	US 0x1F
<u>2</u>	space 0x20	! 0x21	" 0x22	# 0x23	\$ 0x24	% 0x25	& 0x26	' 0x27	(0x28) 0x29	* 0x2A	+ 0x2B	, 0x2C	- 0x2D	. 0x2E	/ 0x2F
<u>3</u>	0 0x30	1 0x31	2 0x32	3 0x33	4 0x34	5 0x35	6 0x36	7 0x37	8 0x38	9 0x39	: 0x3A	; 0x3B	< 0x3C	= 0x3D	> 0x3E	? 0x3F
<u>4</u>	@ 0x40	A 0x41	B 0x42	C 0x43	D 0x44	E 0x45	F 0x46	G 0x47	H 0x48	I 0x49	J 0x4A	K 0x4B	L 0x4C	M 0x4D	N 0x4E	O 0x4F
<u>5</u>	P 0x50	Q 0x51	R 0x52	S 0x53	T 0x54	U 0x55	V 0x56	W 0x57	X 0x58	Y 0x59	Z 0x5A	[0x5B	\ 0x5C] 0x5D	^ 0x5E	_ 0x5F
<u>6</u>	` 0x60	a 0x61	b 0x62	c 0x63	d 0x64	e 0x65	f 0x66	g 0x67	h 0x68	i 0x69	j 0x6A	k 0x6B	l 0x6C	m 0x6D	n 0x6E	o 0x6F
<u>7</u>	p 0x70	q 0x71	r 0x72	s 0x73	t 0x74	u 0x75	v 0x76	w 0x77	x 0x78	y 0x79	z 0x7A	{ 0x7B	 0x7C	} 0x7D	~ 0x7E	DEL 0x7F

☐ Letter
 ☐ Number
 ☐ Punctuation
 ☐ Symbol
 ☐ Other/non-printable

Computers often store variable-length text as a null-terminated string: a sequence of bytes, where each byte denotes a different character, terminated by the value 0x00 (null) to denote the end of the string. For example, the string "6.004" is encoded as the 6-byte sequence 0x362E30303400.

1. Encode your name as a null-terminated ASCII string (use the best approximation if your name contains non-English characters)
2. Decode the following null-terminated ASCII string:

0x 52 49 53 43 2D 56 20 69 73 20 63 6F 6D 69 6E 67 21 00