MSiA-413 Introduction to Databases and Information Retrieval

Homework 1: Number representations

Name 1: Lowan (Sydney) Li

NetID 1: chr0390

Name 2: Ayush Agarwal

NetID 2: scg1143

Instructions

You should submit this homework assignment via Canvas. Acceptable formats are word files, text files, and pdf files. Paper submissions are not allowed and they will receive an automatic zero.

As explained during lecture and in the syllabus, assignments are done in groups. The groups have been created and assigned. Each group needs to submit only one assignment (i.e., there is no need for both partners to submit individually the same homework assignment).

Each group can submit solutions multiple times (for example, you may discover an error in your earlier submission and choose to submit a new solution set). We will grade only the last submission and ignore earlier ones.

Make sure you submit your solutions before the deadline. The policies governing academic integrity, tardiness and penalties are detailed in the syllabus.

Question 1. Unsigned Integer Representation (10 points – 1 point per row)

Please fill out the blank parts of the table below to (i) express the following numbers in binary, (ii) calculate their hexadecimal representation (as shown in class), and (iii) calculate their decimal value as a sum of powers of two.

For example: $52_{10} = 0011\ 0100_2 = 0x34 = 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 0*2^0 = 32 + 16 + 4$

	DECIMAL NUMBER	BINARY	HEXADECIMAL	SUM OF POWERS OF TWO
EXAMPLE	52	0011 0100	0x34	32 + 16 + 4
1A)	6	00000110	0x6	4+2
1B)	19	00010011	0x13	16+2+1
1C)	22	00010110	0x16	16+4+2
1D)	38	00100110	0x26	32+4+2
1E)	42	00101010	0X2A	32+8+2
1F)	155	10011011	0X9B	128+16+8+2+1
1G)	612	1001100100	0X264	512+64+32+4
1H)	1819	11100011011	0X71B	1024+512+256+16+8+2+1
1I)	2293	100011110101	0X8F5	2048+128+64+32+16+4+1
1J)	3176	110001101000	0XC68	2048+1024+64+32+8

Question 2. Signed Integer Representation (10 points – 1 point per row)

Please fill out the blank parts of the table below as needed to calculate the 8-bit two's complement binary representation of the following signed decimal numbers. |x| denotes the absolute value of x. If particular number(s) below cannot be represented as 8-bit signed binary integer(s), please indicate which one(s) and explain why. The explanation will then carry the points of the corresponding row(s). Please remember that:

- You may need to add zeros to the left of the number to make it an 8-bit binary number.
- In two's complement signed integer representation, for every (positive or negative) integer x, it holds: -x = ~x + 1, where ~x is the complement of x (calculated by flipping the bits of x).
- If you don't need a column to calculate the 8-bit signed integer binary representation of a number, you can leave that column blank (as in example 1).

For example: $82_{10} = 0101\ 0010_2$. Note that we added a zero at the front to make it an 8-bit binary number. The sign bit (most significant bit of its 8-bit binary representation) is 0, indicating a positive integer, so the 8-bit signed integer binary representation of 82_{10} is $0101\ 0010_2$.

However, to calculate the 8-bit signed integer representation of -82 we need to calculate its two's complement. We have $|-82_{10}| = 82_{10} = 0101\ 0010_2$. Flipping the bits gives $1010\ 1101_2$. Finally, adding one gives $1010\ 1110_2$. Thus, the 8-bit signed binary representation of -82_{10} is $1010\ 1110_2$.

	VALUE OF INTEGER X	|X| IN 8-BIT BINARY	~|X| IF NEEDED	8-BIT SIGNED INTEGER REPRESENTATION
EXAMPLE 1	82	**$0101\ 0010 = 82_{10}$**		**$0101\ 0010 = 82_{10}$**
EXAMPLE 2	-82	**$0101\ 0010 = 82_{10}$**	**1010 1101**	**$1010\ 1110 = -82_{10}$**
2A)	-1	$00000001 = 1_{10}$	11111110	$11111111 = -1_{10}$
2B)	-19	$00010011 = 19_{10}$	11101100	$11101101 = -19_{10}$
2C)	22	$00010110 = 22_{10}$		$00010110 = 22_{10}$
2D)	-38	$00100110 = 38_{10}$	11011001	$11011010 = -38_{10}$
2E)	-42	$00101010 = 42_{10}$	11010101	$11010110 = -42_{10}$
2F)	68	$01000100 = 68_{10}$		$01000100 = 68_{10}$
2G)	-100	$01100100 = 100_{10}$	10011011	$10011100 = -100_{10}$
2H)	127	$01111111 = 127_{10}$		$01111111 = 127_{10}$
2I)	128	–	–	–
2J)	-129	–	–	–

If you found any numbers that you cannot represent as 8-bit signed integers, indicate which ones and why they cannot be represented in the space below.

Question 3. Binary Pattern Representation (10 points)

BIT PATTERN 1	BIT PATTERN 2
1101 1110 1010 1101 1011 1110 1110 1111	0100 1100 0100 1111 0100 1100 0010 0001

23 19 11

What value does bit pattern 1 represent when interpreted as a

3a) **(2 points)** 32-bit signed integer in two's complement arithmetic?

$$-2^{31} + 2^{30} + 2^{28} + 2^{27} + 2^{26} + 2^{25} + 2^{23} + 2^{21} + 2^{19} + 2^{18} + 2^{16} + 2^{15} + 2^{13} + 2^{12} + 2^{11} + 2^{10} + 2^{9} + 2^{7} + 2^{6} + 2^{5} + 2^{3} + 2^{2} + 2^{1} + 2^{0}$$

$$= -559038737$$

3b) **(2 points)** 32-bit <u>unsigned</u> integer?

$$2^{31} + 2^{30} + \dots$$

$$= 3735928559$$

3c) **(1 point)** write the value in question 3b as a hexadecimal number

DEADBEEF

What value does bit pattern 2 represent when interpreted as a

3d) **(2 points)** 32-bit signed integer in two's complement arithmetic?

$$2^{30} + 2^{27} + 2^{26} + 2^{22} + 2^{19} + 2^{18} + 2^{17} + 2^{16} + 2^{14} + 2^{11} + 2^{10} + 2^{5} + 2^{0}$$

$$= 1280265249$$

3e) **(2 points)** 32-bit <u>unsigned</u> integer?

1280265249

3f) **(1 point)** UTF-8 text?

LOL!

Question 4. Accuracy of Integer and Floating Point Representations (20 points – 1 per row)

In the table below, answer "***exactly***," "***approximately***," or "***no way***" to indicate how the following (base ten) numbers can be represented as 32-bit signed integers, 64-bit signed integers, single precision (32-bit) floats, and double precision (64-bit) floats. If the number can be rounded to a value that is representable and the relative error is less than 10^{-3}, then the best answer is "approximately." For example, I would say that 2.5 is "no way" 3, but 2.9999999 is "approximately" 3. The relative error definition is in the reading material and was explained in the lab session.

For example, consider the number $1.0000000005 \times 10^{30}$. It cannot be represented as 32-bit signed integer nor as a 64-bit signed integer (because it is too large). However, it can be represented approximately as a single precision float (with rounding – there are not enough bits in the mantissa of the 32-bit float to represent the 0.0000000004 fractional part of the number, but the relative error is $<10^{-3}$). Moreover, it can be represented exactly as a double precision float.

You can use online floating-point conversion tools (e.g., http://www.binaryconvert.com/convert_float.html).

	BASE TEN NUMBER	32-BIT *SIGNED* INTEGER	64-BIT *SIGNED* INTEGER	32-BIT FLOATING POINT	64-BIT FLOATING POINT
EXAMPLE	1.0×10^{20}	**no way**	**no way**	**approximately**	**exactly**
4A	-0.5	N	N	E	E
4B	1/3 (one third)	N	N	A	A
4C	0.1	N	N	A	A
4D	1/16 (one sixteenth)	N	N	E	E
4E	17.5	N	N	E	E
4F	4,000,000,000	N	E	E	E
4G	-4,000,000,001	N	E	A	E
4H	$\sqrt{2}$	N	N	A	A
4I	2,000,000,000	E	E	E	E
4J	2,000,000,001	E	E	A	E
4K	20,000,000	E	E	E	E
4L	20,000,000.25	A	A	A	E
4M	33,554,432.25	A	A	A	E
4N	33,554,432.9	A	A	A	A
4O	9,123,000,000,000,000,000	N	E	A	E
4P	-9,123,123,123,000,000,000	N	E	A	A
4Q	9,123,123,123,123,123,123	N	E	A	A
4R	pi	N	N	A	A
4S	infinity	N	N	E	E
4T	zero	E	E	E	E