

Practice 2 (2016/9/20)
Friendly and Smart Interactive Query – (I)

1. Binary Number System

A. Power of 2

Exponent	Value
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

Exponent	Value
11	2,048
12	4,096
13	8,192
14	16,384
15	32,768
16	65,536
17	131,072
18	262,144
19	524,288
20	1,048,576
21	2,097,152

- 2^{10} (1024) is Kilo, denoted "K"
- 2^{20} (1,048,576) is Mega, denoted "M"
- 2^{30} (1,073, 741,824) is Giga, denoted "G"

☞ All data are stored in a computer as binary numbers in different formats, such as signed, unsigned, floating point (IEEE Std 754), etc.

B. Conversion between binary number and decimal number

$$11010_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 = 26_{10}$$

$$26/2 = 13 + (0/2)$$

$$13/2 = 6 + (1/2)$$

$$6/2 = 3 + (0/2)$$

$$3/2 = 1 + (1/2) \rightarrow 11010_2$$

C. Negative binary number

☞ Signed Magnitude: 0000 0000 0000 0001 vs. 1000 0000 0000 0001

☞ For an n-bit number, maximum positive number = $2^{(n-1)} - 1$;

minimum negative number = $-(2^{(n-1)} - 1)$;

☞ 2's Complement: 0000 0000 0000 1001 vs. 1111 1111 1111 0111

$$☞ 2 - 1 = 2 + (-1) = 00000010_2 + 11111111_2 = 00000001_2 = 1_{10}$$

☞ For an n-bit number, maximum positive number = $2^{(n-1)} - 1$;

minimum negative number = $-2^{(n-1)}$;

2. Octal and Hexadecimal Number

A. Octal digits and hexadecimal digits

☞ 0 ~ 7 for Octal and {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F} for Hexadecimal.

☞ One Octal digit is composed of **three binary bits** while **four binary bits** for **one hexadecimal digit**.

B. Conversion among binary, octal, and hexadecimal numbers

Decimal (Base 10)	Binary (Base 2)	Octal (Base 8)	Hexadecimal (Base 16)
00	00000	00	00
01	00001	01	01
02	00010	02	02
03	00011	03	03
04	00100	04	04
05	00101	05	05
06	00110	06	06
07	00111	07	07
08	01000	10	08
09	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10

☞ **Tip:** Convert any non-binary number to binary number first, then you can easily convert it to another based number.

☞ **Example:** $26_{10} = 11010_2 = (011)(010)_2 = 32_0 = (0001)(1010)_2 = 1A_H$

☞ Try to print positive and negative numbers in octal and hexadecimal format.

☞ Which type of signed representation does C use to store a negative number?

3. American Standard Code for Information Interchange (ASCII)

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

☞ All characters are stored as numbers too.

☞ `char newCh = 'y';`

4. Conditional Statement and Dishonored Statement

☞ `if (newCh == 121) { b++; c--; }` ☞

`else {b--; c++;}`

☞ Other relations: `>`, `<`, `>=`, `<=`, `!=`

☞ Boolean relation: `&&` (and), `||` (or); priority: `&&` `>` `||`

`if (5 > 1 || 1 > 2 && 3 < 1) printf("and > or\n");`

☞ `if ('a' < 'b') → True; if ('A' > 'a') → False`

☞ Dishonored statement goto:

`int c;`

`Label1: scanf("%d", &c);`

`statements;`

`if (c < 10) goto Label1; /* input a number that is not less than 10 */`

`...`

5. More Detail of scanf()

A. Keyboard buffer

Keyboard Buffer



B. What are the return values of `scanf("%d %d" ...)` and `scanf("%c %d" ...)`?

6. Problem

When you want users to confirm one thing, you usually ask users to input 'y' (Yes) or 'n' (No) no matter if it is uppercase or lowercase. Write a program to complete this query. You have to check if the input character is 'y', 'Y', 'n', or 'N'. If it is not the case, the program has to repeat the query until the input character matches your desired input. A friendly program will remind users what wrong it is after users input an invalid character. And then enter the query procedure again. To show your program is somewhat smart too, tell users what kind of character their input character is if their input character is invalid. For instance, if users input a '1', the program can display a message like this "Your input is a number character. Please input 'y' or 'n' but not a number character.". Your message can be scenario-dependent. Users will have a warming experience rather than cold using experience.

Main features:

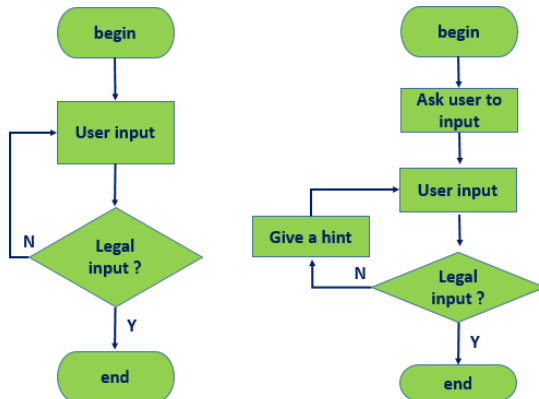
(a) Repeat query if input character is invalid.

(b) Tell users what kind of character their input character is and remind them of

valid range for their input.

☞ At least complete (a) before the end of this practice time.

☞ The deadline to upload the program with all features is the midnight of 9/26.



7. Bonus

Modify your program to read in two integers separated by a space. You can check the return value of `scanf()` to see how many integers it really receives. A user may intentionally or accidentally input non-numerical character. Try to refine your program to contain repeated query for receiving two integers until you really input two integers. Just compile and run your program to see if the program really runs as your anticipation. If yes, congratulations to your achievement. Remember to mail me your program and tell me what you have done for this modified version. If not, don't feel frustrated. Write some test programs and try to find out the possible reasons under your deep insight. Mail me your test programs and tell me your analysis and observations.

☞ Test your program by invalid inputs such as the combinations of a character and an integer or just two characters. E.g., "a 10" or "a b".

☞ The deadline of Bonus is still the midnight.

☞ You have to finish features (a) and (b) first and submit your program to E3.

After that, you are qualified to take this challenge.

8. Further Reading

A. `getchar()`, conditional compilation, IEEE Std 754