

1. Number Conversion

Fill in the table below

Binary	Octal	Decimal	Hexadecimal
110 1100	154	108	0x6C
110 0101 1101	3135	1629	0x65D
1011 1110 1101	5755	3053	0xBED
111 1101 1110	3736	2014	0x7DE
100 0000 1100	2014	1036	0x40C

2. Operations

(1) Given A and B with hexadecimal expression 0xED and 0x3F respectively. Calculate the values of the following expressions

- a) A & B 0x2D
- b) A && B 1
- c) A | B 0xFF
- d) A || B 1
- e) A ^ B 0xD2
- f) ~A | ~B 0xD2

(2) Fill in the table below with the results of shift operation given below (Assume X is 8-bit wide) . Please answer in hexadecimal.

X	X<<4	X>>3(Logical)	X>>3(Arithmetic)
0x5D	0xD0	0x0B	0x0B
0xC9	0x90	0x19	0xF9
0xA2	0x20	0x14	0xF4
0xBF	0xF0	0x17	0xF7

(3) Design C expressions using bitwise (& | ~ ^) and logical (&& || !) operators, which return 1 under the specific conditions described below, otherwise return 0.

- A. all bits of x are 1. !(~x)
- B. all bits of x are 0. !x
- C. bits of x's least significant byte are 1. !(~x & 0xFF)
- D. bits of x's least significant byte are 0. !(x & 0xFF)

(4) Design a C expression, which generates a word (32-bit) consisting of the lower 16 bits of x and the remaining bits of y.

For example, x = 0x89ABCDEF and y = 0x76543210, it will generate 0x7654CDEF

(x & 0xFFFF) | (y & ~0xFFFF)

3. Byte Ordering

(1) Consider the following function:

```
typedef unsigned char *byte_pointer;
```

```

void show_bytes(byte_pointer start, int len)
{
    int i;
    for (i = 0; i < len; i++)
        printf("%.2x", start[i]);
    printf("\n");
}

```

```

int val = 0x28AE3514
byte_pointer valp = (byte_pointer) &val;

```

What is the output of the following call to show_bytes on big-endian and little-endian machines respectively?

	little-endian	big-endian
show_bytes(valp, 1);	14	28
show_bytes(valp, 2);	14 35	28 AE
show_bytes(valp, 3);	14 35 AE	28 AE 35
show_bytes(valp, 4);	14 35 AE 28	28 AE 35 14