

Student ID: PB22111679

Name: 孙婧雯

Course Title: ICS

Homework 1

T1

1. $-114(D) = 11110010(B)$, so its complement number is 10001110 .
 $+81(D) = 01010001(B)$, so its complement number is 01010001 .
2. The original number of 00110010 is 00110010 , namely $2^1 + 2^4 + 2^5 = +50(H)$.
The original number of 11111101 is 10000011 , namely $-3(H)$.

T2

1. The largest number is $+127$, whose complement number is 01111111 . The smallest number is -128 , whose complement number and itself are both 10000000 .
2. The number that can be represented by an N -bit 2's complement number ranges from $-2^{(N-1)}$ to $+2^{(N-1)}+1$, $2N$ numbers in total.

T3

Assume that number is $\overline{1X}$ in 8-bit 2's complement number.

Then

$$X = 1 + \bar{X}$$

solve that

$$X = 0000000$$

Then the answer must be 10000000 , namely -128 in decimal.

T4

1. If $0 \leq a \leq 2^{31} - 1$ and $b = -2^{31}$ at the same time, the program will print $a < b$.

When the computer is doing subtraction, it's actually doing addition by complement number. So any non-negative number minus b will produce a '1' in front of the result, which shows that 'a - b < 0'.

2. If we change the code, the wrong answer in T4-1 will be corrected.

T5

0 10001011 00000000001000000001000

The sign bit is 0, signifying a positive number. The next 8 bits is 139, signifying an exponent of $139 - 127 = +12$. So we actually get $1 * 2^{12} * (2^{-11} + 2^{-20}) = 2.00390625$.

T6

The smallest number is 1 11111110 111111111111111111111111, which is approximately -3.4e38.

The smallest positive number is the smallest subnormal

0 00000000 000000000000000000000001, which is approximately 1.4e-45.

T7

Write a program to check if the 2's complement number and IEEE floating point if a certain integer (range from -2147483648 to 2147483647) meet our need.

Use forced cast type and binary operation to get the complement number.

The answers are -834214802, 0 and 1318926965.

 输出

```
1 -834214802
2 0
3 1318926965
4
```

T8

1. $a = a \oplus b;$
 $b = a \oplus b;$
 $a = a \oplus b;$

2. Yes.

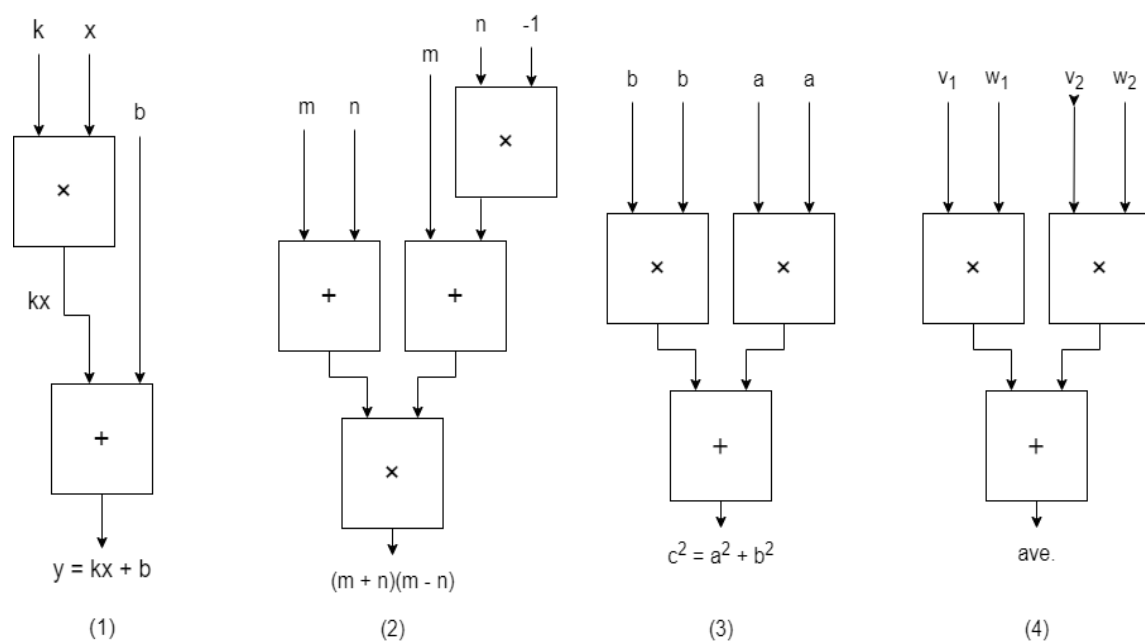
If we cannot find a j , then $a[i]$ will XOR itself and become 0;

Fix:

```
void sort(int *a, int n) {  
    // sort a[0] ~ a[n - 1]  
    for (int i = 0; i < n - 1; i++) {  
        int min = i, flag = 0;  
        for (int j = i; j < n; j++) {  
            if (a[j] < a[min]) {  
                min = j;  
                flag = 1;  
            }  
        }  
        if(flag) swap(a + i, a + min);  
    }  
}
```

T9

The circuits are as follows.



T10

1. There are $26 + 26 + 10 = 64$ characters in total. To avoid crash or error, we need just $\log_2(64) = 6$ bits to represent a single character.

2. $6 \cdot N$ bits

3. The decimal representation of Hello world. is 8 30 37 37 40 62 58 40 43 37 29 63. So the binary representation is
001000 011110 100101 100101 101000 111110
111010 101000 101011 100101 011101 111111