The University of Hong Kong
Department of Computer Science
COMP2120 A/B Computer organization

1. (a) i. (2 points) Write down the 8-bit two's complement representation of the numbers 37 and -24.

> **Solution:** 37: 00100101, -24: 11101000

ii. (3 points) Write down the mapping function f for converting a bit pattern P (represented as an unsigned integer) to the represented value V in both two's complement representation and excess-k representation. $V = f(P)$

> **Solution:** Two's complement: $V = P - 2^{n-1}$(if V is negative), Excess-k: $V = P - k$

(b) Consider a 32-bit floating point representation with a sign bit S, an exponent E (biased, 12 bits), and a significand f (19 bits). No need to consider IEEE special patterns. The value is

S	12-bit exponent E	19-bit significand f

i. (1 point) Write down the value of b.

Solution: $b = 2^{12-1} - 1 = 2047$

ii. (1 point) Write down the largest positive number that can be represented.

Solution: When $S = 0, E = 111\ldots11, f = 111\ldots11$,
V is the largest positive number that can be represented: $2^{2049} - 2^{2029}$.

iii. (3 points) Write down the bit pattern representing the value 30.625

Solution: $30.625_{10} = 11110.101_2 = 1.1110101 \times 10^4$
Thus, $f = 1110\ 1010\ 0000\ \ldots 0000$, $E = 2051 = 1000\ 0000\ 0011$, $S = 0$.
The bit pattern is 0100 0000 0001 1111 0101 0000 0000 0000 = `401F5000`.

2. (a) (3 points) Consider the following function, `a[][]` are stired in memory with the following addresses.

```
int i, j, sum = 0;
    for (j = 0; j < M; j++)
        for (i = 0; i < N; i++)
            sum += a[i][j];
return sum;
```

Address	0	4	8	12	16	20
Contents	a_{00}	a_{01}	a_{02}	a_{10}	a_{11}	a_{12}

Please write down the access order of `a[][]` in the function. Please modify the function to have a better spatial locality.

Solution: Access Order: a_{00}, a_{10}, a_{01}, a_{11}, a_{02}, a_{12}.
Modified Function:

```
int i, j, sum = 0;
for (i = 0; i < N; i++)
    for (j = 0; j < N; j++)
        sum += a[i][j];
return sum;
```

(b) Consider a hypothetical machine with 1024 words of cache memory. They are in a two-way set associative organization, with cache block size of 128 words. One word is four bytes.

i. (1 point) How many blocks are there in the cache memory?

Solution: #blocks $= 1024/128 = 8$

ii. (1 point) How many sets are there in the cache memory?

Solution: #sets $= 8/2 = 4$

iii. (5 points) The memory address is byte addressable, i.e. each address is one byte. For the byte with address `C912F`(hex), what are its cache set number, cache tag, and offset? Suppose this byte is stored in the cache. What are the addresses of the other bytes stored along with it?

Solution: `C912F` = 1100 1001 0001 0010 1111

offset	$log_2(128 \times 4) = 9$ bits
cache set	$\log_2(4) = 2$ bits
cache tag	$20 - 9 - 2 = 9$ bits

Thus, the cache set number is 00, the cache tag is 110010010 and the offset is 100101111. The addresses of the other bytes stored along with it are from 1100 1001 0000 0000 0000 to 1100 1001 0001 1111 1111, namely, from `C9000` to `C91FF` (hex).