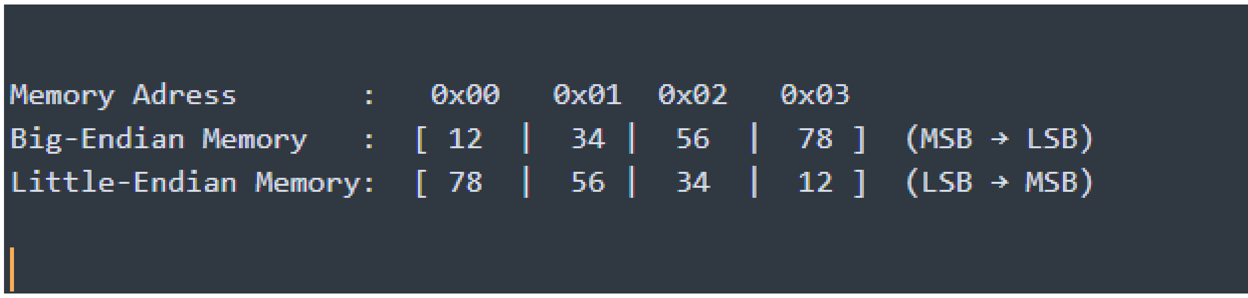
1. What are the rules of big-endian and little-endian? Illustration
    a. Big-Endian Rules
        i. The **most significant byte (MSB)** is stored **first** (at the lowest memory address)
        ii. The **least significant byte (LSB)** is stored **last** (at the highest memory address).

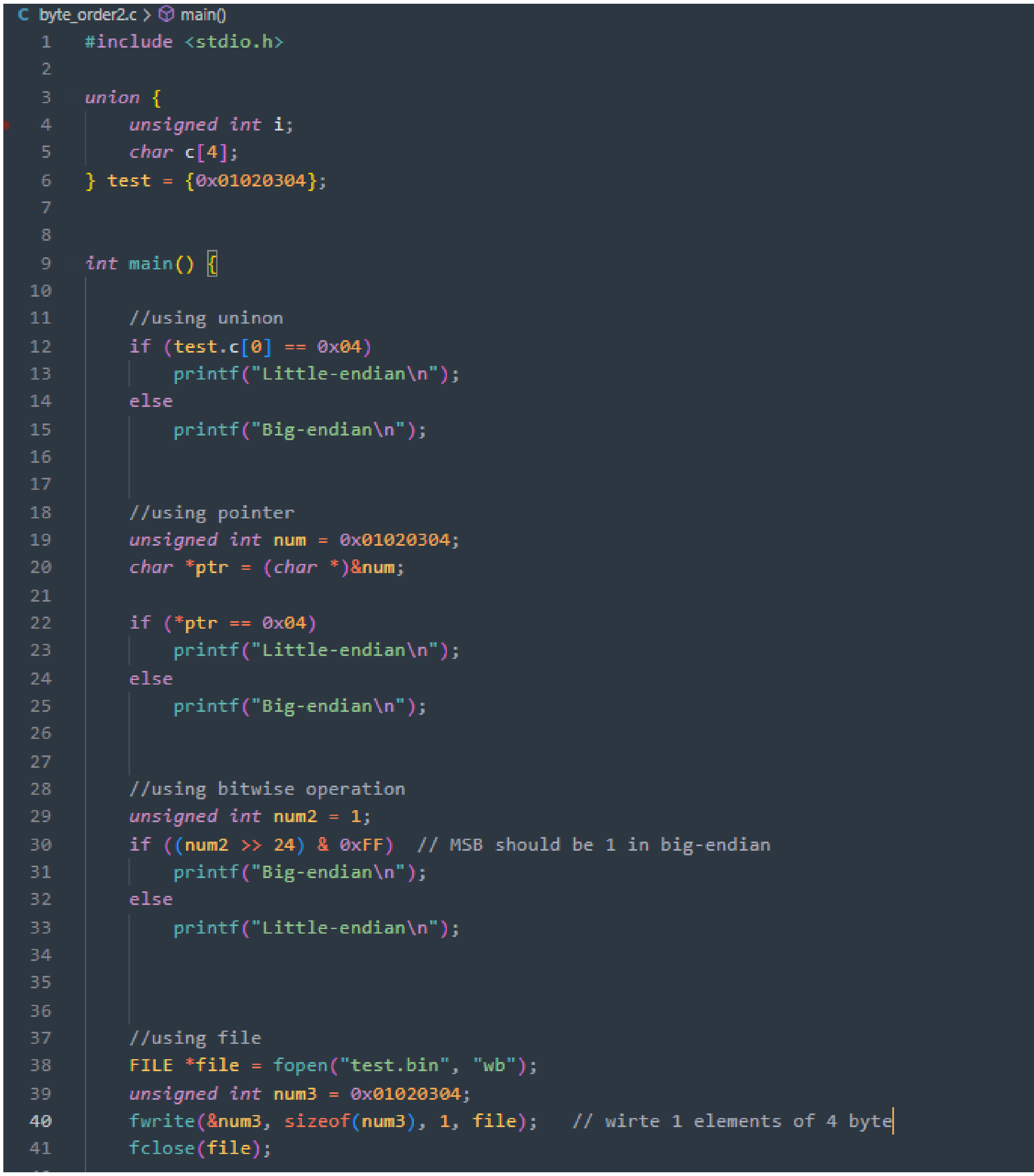
    b. Little-Endian Rules
        i. The **least significant byte (LSB)** is stored **first** (at the lowest memory address).
        ii. The **most significant byte (MSB)** is stored **last** (at the highest memory address).

    c. **Illustration**: storing 0x12345678 (4-byte integer)

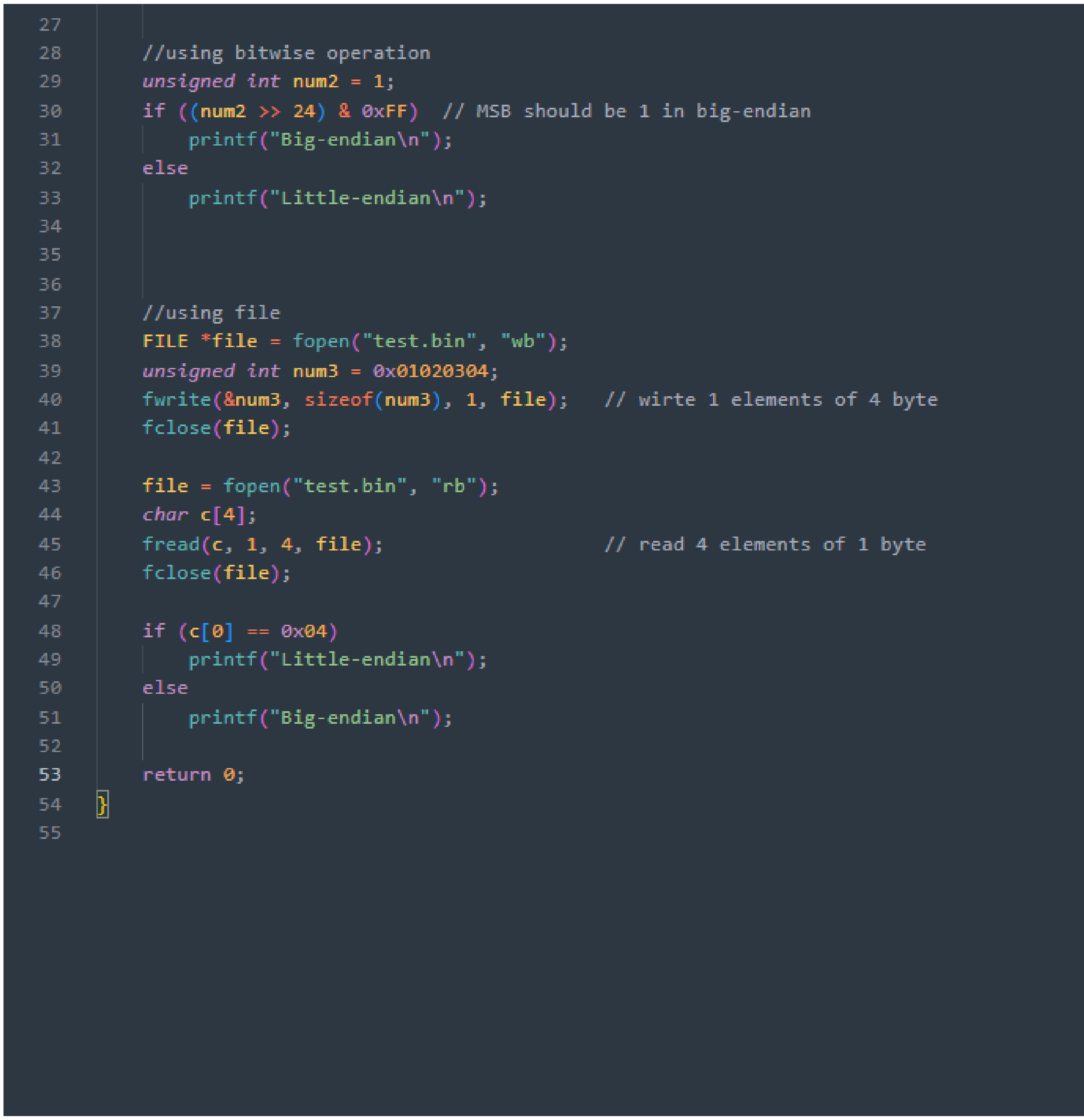


```
Memory Adress       :   0x00   0x01  0x02   0x03
Big-Endian Memory   :  [ 12  |  34 |  56  |  78 ]  (MSB → LSB)
Little-Endian Memory:  [ 78  |  56 |  34  |  12 ]  (LSB → MSB)
```

2. How many ways to determine the byte order of your PC? Illustration
    a. There are many ways to determine the byte order in my pc. Some illustration given below.


C byte_order2.c > main()

```c
#include <stdio.h>

union {
    unsigned int i;
    char c[4];
} test = {0x01020304};


int main() {

    //using uninon
    if (test.c[0] == 0x04)
        printf("Little-endian\n");
    else
        printf("Big-endian\n");


    //using pointer
    unsigned int num = 0x01020304;
    char *ptr = (char *)&num;

    if (*ptr == 0x04)
        printf("Little-endian\n");
    else
        printf("Big-endian\n");


    //using bitwise operation
    unsigned int num2 = 1;
    if ((num2 >> 24) & 0xFF)  // MSB should be 1 in big-endian
        printf("Big-endian\n");
    else
        printf("Little-endian\n");



    //using file
    FILE *file = fopen("test.bin", "wb");
    unsigned int num3 = 0x01020304;
    fwrite(&num3, sizeof(num3), 1, file);   // wirte 1 elements of 4 byte
    fclose(file);
```



```c

    //using bitwise operation
    unsigned int num2 = 1;
    if ((num2 >> 24) & 0xFF)  // MSB should be 1 in big-endian
        printf("Big-endian\n");
    else
        printf("Little-endian\n");



    //using file
    FILE *file = fopen("test.bin", "wb");
    unsigned int num3 = 0x01020304;
    fwrite(&num3, sizeof(num3), 1, file);   // wirte 1 elements of 4 byte
    fclose(file);

    file = fopen("test.bin", "rb");
    char c[4];
    fread(c, 1, 4, file);                   // read 4 elements of 1 byte
    fclose(file);

    if (c[0] == 0x04)
        printf("Little-endian\n");
    else
        printf("Big-endian\n");

    return 0;
}

```

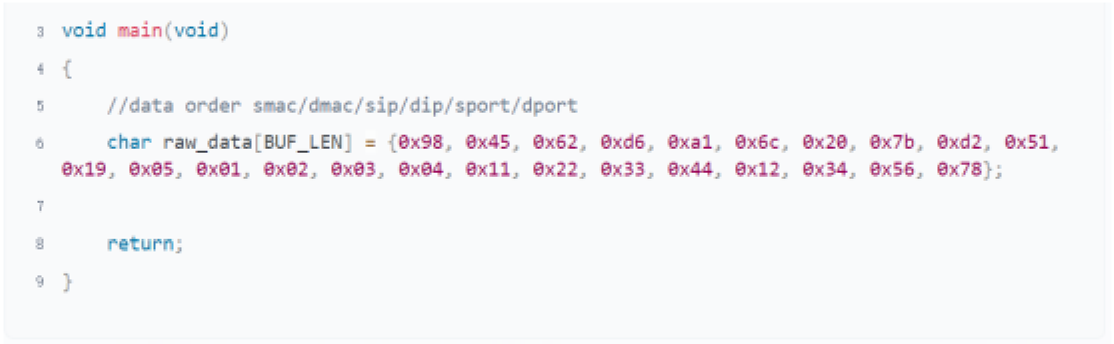
3. Computer-Systems-A-Programmers-Perspective.pdf Problem 2.6
   Using show_int and show_float, we determine that the integer 2607352 has hexadecimal representation 0x0027C8F8, while the floating-point number 3510593.0 has hexadecimal representation 0x4A1F23E0.
   A. Write the binary representations of these two hexadecimal values.
   B. Shift these two strings relative to one another to maximize the number of matching bits. How many bits match?
   C. What parts of the strings do not match?
   a.
      0 0 2 7 7 C F 8
      0000 0000 0010 0111 1100 1000 1111 1000

      4 A 1 F 2 3 E 0
      0100 1010 0001 1111 0010 0011 1110 0000

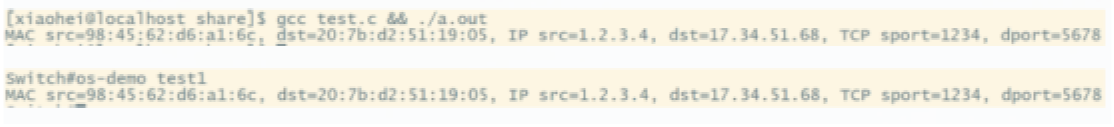
   b. shifting two position left relative to the second or shifting two position right relative to the first 21 matching bits.
      0000 0000 0010 0111 1100 1000 1111 1000
      0001 0010 1000 0111 1100 1000 1111 1000

   c. most significants bits of integer doesn’t match with the floating point number and some of higher order bits of floating points number doesn’t match with the integer.

4. Here is a bunch of data from the network(big-endian). Parse and print it following the format as the image blew to ensure it can run properly;

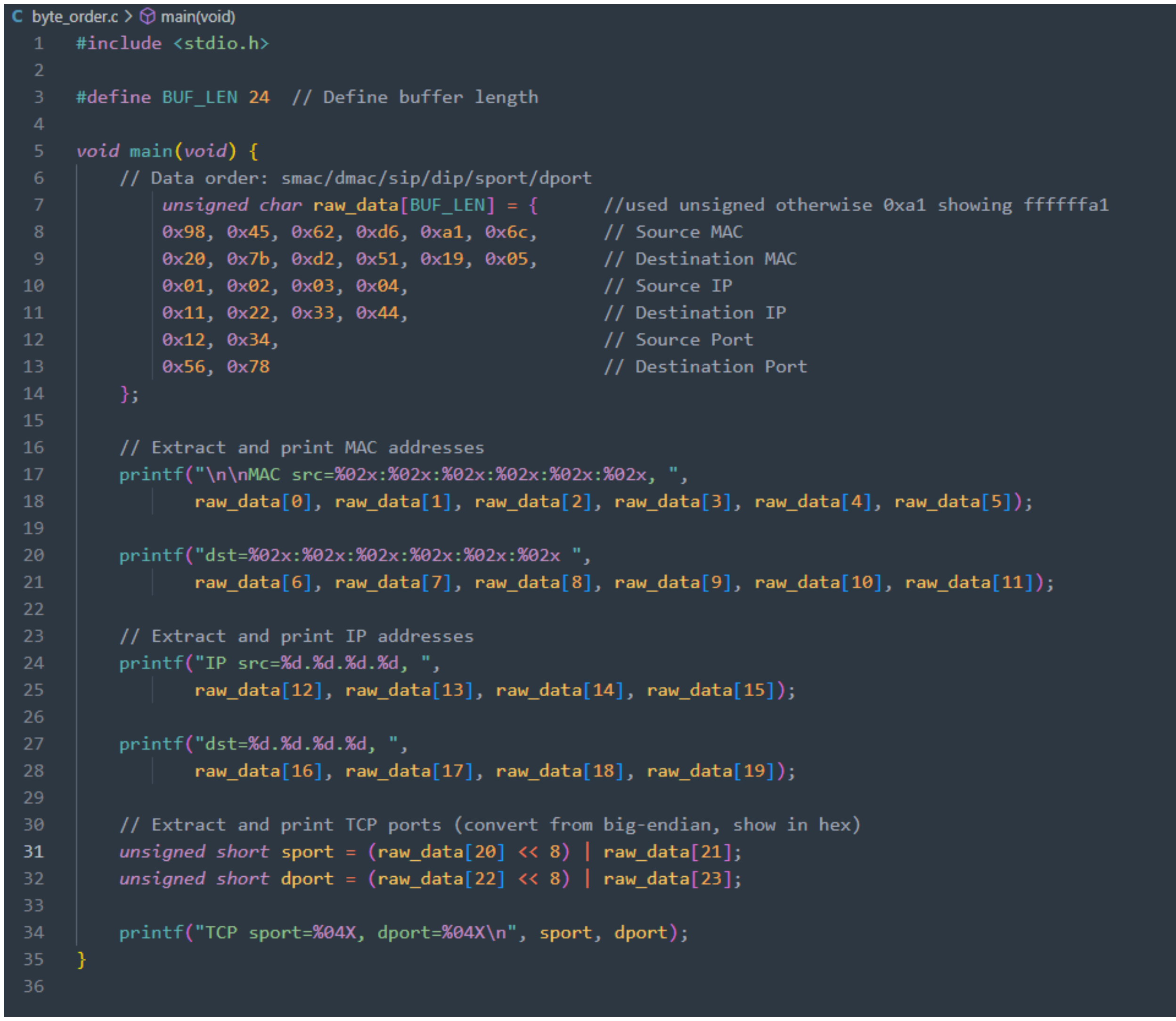


```
void main(void)
{
    //data order smac/dmac/sip/dip/sport/dport
    char raw_data[BUF_LEN] = {0x98, 0x45, 0x62, 0xd6, 0xa1, 0x6c, 0x20, 0x7b, 0xd2, 0x51,
0x19, 0x05, 0x01, 0x02, 0x03, 0x04, 0x11, 0x22, 0x33, 0x44, 0x12, 0x34, 0x56, 0x78};

    return;
}
```



```
[xiaohei@localhost share]$ gcc test.c && ./a.out
MAC src=98:45:62:d6:a1:6c, dst=20:7b:d2:51:19:05, IP src=1.2.3.4, dst=17.34.51.68, TCP sport=1234, dport=5678

Switch#os-demo test1
MAC src=98:45:62:d6:a1:6c, dst=20:7b:d2:51:19:05, IP src=1.2.3.4, dst=17.34.51.68, TCP sport=1234, dport=5678
```

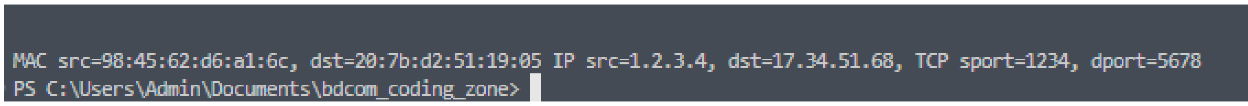
   a. code:



byte_order.c > main(void)

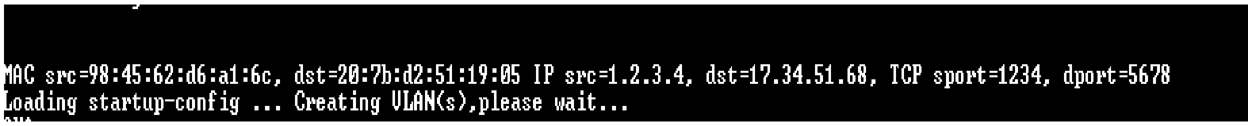
```c
#include <stdio.h>

#define BUF_LEN 24  // Define buffer length

void main(void) {
    // Data order: smac/dmac/sip/dip/sport/dport
        unsigned char raw_data[BUF_LEN] = {      //used unsigned otherwise 0xa1 showing ffffffa1
        0x98, 0x45, 0x62, 0xd6, 0xa1, 0x6c,      // Source MAC
        0x20, 0x7b, 0xd2, 0x51, 0x19, 0x05,      // Destination MAC
        0x01, 0x02, 0x03, 0x04,                  // Source IP
        0x11, 0x22, 0x33, 0x44,                  // Destination IP
        0x12, 0x34,                              // Source Port
        0x56, 0x78                               // Destination Port
    };

    // Extract and print MAC addresses
    printf("\n\nMAC src=%02x:%02x:%02x:%02x:%02x:%02x, ",
           raw_data[0], raw_data[1], raw_data[2], raw_data[3], raw_data[4], raw_data[5]);

    printf("dst=%02x:%02x:%02x:%02x:%02x:%02x ",
           raw_data[6], raw_data[7], raw_data[8], raw_data[9], raw_data[10], raw_data[11]);

    // Extract and print IP addresses
    printf("IP src=%d.%d.%d.%d, ",
           raw_data[12], raw_data[13], raw_data[14], raw_data[15]);

    printf("dst=%d.%d.%d.%d, ",
           raw_data[16], raw_data[17], raw_data[18], raw_data[19]);

    // Extract and print TCP ports (convert from big-endian, show in hex)
    unsigned short sport = (raw_data[20] << 8) | raw_data[21];
    unsigned short dport = (raw_data[22] << 8) | raw_data[23];

    printf("TCP sport=%04X, dport=%04X\n", sport, dport);
}
```

b. output in vscode:



```
MAC src=98:45:62:d6:a1:6c, dst=20:7b:d2:51:19:05 IP src=1.2.3.4, dst=17.34.51.68, TCP sport=1234, dport=5678
PS C:\Users\Admin\Documents\bdcom_coding_zone>
```

c. output in switch



```
MAC src=98:45:62:d6:a1:6c, dst=20:7b:d2:51:19:05 IP src=1.2.3.4, dst=17.34.51.68, TCP sport=1234, dport=5678
Loading startup-config ... Creating VLAN(s),please wait...
```