

1. Consider the following C code running in a RISC-V environment. What will be printed to the standard output? Please explain your answer. (Hint: **endianness** of RISC-V processors)

```
#include <stdint.h>
#include <stdio.h>

int main() {
    char mem[] = {0xFE, 0xDC, 0xBA, 0x98};
    uint32_t result = *(uint32_t *)mem;
    printf("%x", result);
}
```

1. 98BADCFE 或 98badcfe , 0x98BADCFE 不扣分
2. 提到 little-endian 或 small endian 就給分，或是最高位元組存在高記憶體位置也給分。

「Least endian」、「小印地安」不給分，「最高位在右邊」的敘述也不給分。

2. The IEEE Standard for Floating-Point Arithmetic (IEEE 754) is a technical standard for floating-point arithmetic established in 1985 by the Institute of Electrical and Electronics Engineers (IEEE). This standard addresses many problems found in the diverse floating-point implementations that made them difficult to use reliably and portably. Many processors conform to the IEEE 754 standard for representing the floating-point numbers. For example, the bit sequence of -0.75 in a 32-bit register based on IEEE754 is 1011111101000...00. Please write down the **hexadecimal strings** of -0.125 and 0.1875, respectively, using the IEEE 754 standard to represent floating-point data in a 32-bit register.

1. 0xBE000000
2. 0x3E400000

題目已經明示且粗體需要寫 16 進位，寫 2 進位不給分