Task 1:

Submitted By: Sagar More

Write proper structure implementation for all RCC registers and bit field implementation AHBENR1 register bits and structure and bit field you have to do for IDR and ODR and MODE register and GIPO regarding

#Main.c

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

uint32_t RegVal = 0xFE3C9AD7;

// Normal structure (manual bit extraction)
struct node {
    uint8_t flag; // 1 bit
    uint8_t temp; // 2 bits
    uint8_t mac;  // 4 bits
    uint16_t crc; // 12 bits
    uint8_t bat;  // 4 bits
    uint8_t sens; // 3 bits
    uint8_t air;  // 2 bits
};

// Bitfield structure
struct Node {
    uint32_t flag : 1;
    uint32_t temp : 2;
    uint32_t mac  : 4;
    uint32_t crc  : 12;
    uint32_t bat  : 4;
    uint32_t sens : 3;
    uint32_t air  : 2;
    uint32_t      : 4; // padding (optional)
};

// Union for bitfield + 32-bit register
union Register {
    uint32_t all;
    struct Node bits;
};

int main(void)
{
    printf("Bitfield for: 0x%08X\n\n", RegVal);

    /* ------- Without Bitfield ------- */
    struct node n;
    n.flag =  (RegVal >> 0) & 0x1;
    n.temp =  (RegVal >> 1) & 0x3;
    n.mac  =  (RegVal >> 3) & 0xF;
    n.crc  =  (RegVal >> 7) & 0xFFF;
    n.bat  =  (RegVal >> 19) & 0xF;
    n.sens =  (RegVal >> 23) & 0x7;
    n.air  =  (RegVal >> 26) & 0x3;
```

```
    printf("---- Without Bitfield ----\n");
    printf("Sizeof struct: %zu bytes\n", sizeof(n));
    printf("flag = 0x%X\n", n.flag);
    printf("temp = 0x%X\n", n.temp);
    printf("mac  = 0x%X\n", n.mac);
    printf("crc  = 0x%X\n", n.crc);
    printf("bat  = 0x%X\n", n.bat);
    printf("sens = 0x%X\n", n.sens);
    printf("air  = 0x%X\n\n", n.air);

    /* ------- With Bitfield (using union) ------- */
    union Register reg;
    reg.all = RegVal;

    printf("---- With Bitfield (via union) ----\n");
    printf("Sizeof struct: %zu bytes\n", sizeof(reg.bits));
    printf("flag = 0x%X\n", reg.bits.flag);
    printf("temp = 0x%X\n", reg.bits.temp);
    printf("mac  = 0x%X\n", reg.bits.mac);
    printf("crc  = 0x%X\n", reg.bits.crc);
    printf("bat  = 0x%X\n", reg.bits.bat);
    printf("sens = 0x%X\n", reg.bits.sens);
    printf("air  = 0x%X\n", reg.bits.air);

    return 0;
}
```

Output:-

```
PS D:\VS code\C Programming> gcc .\bitfield.c -o bitfield
PS D:\VS code\C Programming> .\bitfield.exe
Bitfield for: 0xFE3C9AD7

---- Without Bitfield ----
Sizeof struct: 10 bytes
flag = 0x1
temp = 0x3
mac  = 0xA
crc  = 0x935
bat  = 0x7
sens = 0x4
air  = 0x3

---- With Bitfield (via union) ----
Sizeof struct: 4 bytes
flag = 0x1
temp = 0x3
mac  = 0xA
crc  = 0x935
bat  = 0x7
sens = 0x4
air  = 0x3
PS D:\VS code\C Programming>
```