C 语言课程.QA

Q1: char buffer[100] 分配的是什么？
A: 它分配的是连续的 100 个字节的内存空间。char 在 C 语言中占 1 字节，所以 buffer 的大小与“字符数量”无关，而与“字节数”有关。

Q2: UTF-8 编码下，英文和中文占用的字节一样吗？
A: 不一样。UTF-8 是变长编码：英文字符（ASCII 字符）通常占 1 字节, 常见中文汉字一般占 3 字节, 有些特殊符号或 Emoji 可能占 4 字节

Q3: 如果把内容写超过 buffer 的容量，会发生什么？
A: 会造成“缓冲区溢出”，属于未定义行为，可能出现以下情况：

1. 覆盖相邻内存的数据
2. 修改其他变量的值
3. 程序崩溃（如段错误）
4. 数据异常但程序继续运行
5. 造成安全隐患（比如被攻击利用）

Q4: 以后如果想按“字符数量”来设计空间怎么办？
A: 要把字符数量换算成字节数。例如要容纳 20 个中文字符（UTF-8），可以写：char buffer[20 * 3 + 1]; 多出的 1 字节用作结束符

Q5: 为什么 C 语言项目（如 1.c 和 2.c）中只能有一个 main() 函数？
A: 这是由 C 语言程序的编译和链接机制决定的。main() 函数被操作系统和链接器视为程序的唯一入口点（Entry Point）

Q6: 如果我在两个文件中都定义了 main()，会发生什么？
A: 会导致链接错误（Linker Error），通常提示为“multiple definition of ‘main’”（多重定义）。

Q7: 为什么链接器会报错？
A: 链接器的任务是将所有目标文件（.o）和库文件组合成一个可执行文件。它必须确定一个唯一的起始地址来开始执行程序。如果找到两个 main 符号，链接器不知道从哪个开始，因此无法完成工作。

Q8: 在多文件 C 项目中，正确的做法是什么？

A: 只在一个文件中（如 main.c）定义 main() 函数作为入口。其他文件（如 util.c）只定义普通的功能函数，并通过头文件（.h）提供函数声明。

```c
// 文件名: util.h

#ifndef UTIL_H
#define UTIL_H

/**
* @brief 计算两个整数的和。
* * @param a 第一个整数
* @param b 第二个整数
* @return int 两个整数的和
*/
int add(int a, int b);

#endif // UTIL_H
```

Listing 1: 头文件：util.h (提供函数声明)

```c
// 文件名: util.c

// 包含自己的头文件，以确保函数定义与声明一致
#include "util.h"

// 函数的具体实现 (定义)
int add(int a, int b) {
    return a + b;
}

// 注意: 这个文件里没有 main() 函数!
```

Listing 2: 功能实现文件：util.c (提供函数定义)

```c
// 文件名: main.c

#include <stdio.h>
#include "util.h" // 引入 util.h, 从而可以使用 add 函数的声明

int main() {
    int num1 = 10;
    int num2 = 5;\part{title}

    // 调用 util.c 中实现的 add 函数
    int sum = add(num1, num2);

    printf("数字 %d 和 %d 的和是: %d\n", num1, num2, sum);

    // main 函数是程序的起点, 同时也是终点
    return 0;
}
```

Listing 3: main.c (包含 main() 函数)

Q9: int main(void) 里面的 void 是什么意思?

A: main 函数后面的 (void) 表示 main 函数不接受任何参数, 空括号 () 等价于 int main(void)，也表示不接受参数

Q10: 什么是加减法交换算法

A: 利用变量 A 来临时存储 A 和 B 的总和，然后通过减法运算，从这个总和中“分离”出原始的 A 和 B 的值。

Q11: 为什么不使用临时变量更难?

A: 通常情况下，我们使用一个临时变量 temp 来交换, 这种方法最清晰、最安全。加减法交换的优势在于节省内存：避免了声明和使用额外的 temp 变量所占用的内存空间（虽然在现代计算机中这几乎可以忽略不计）

```c
int a = 5, b = 10;
int temp; // 临时变量
temp = a;  // temp = 5
a = b;     // a = 10
b = temp;  // b = 5 内容...
```

Listing 4: 代码如下

Q12: 什么是常量

A: 常量是程序运行时值不能改变的量。在 C 语言中，定义常量主要有两种方法, 使用 const 关键字, 使用 define 预处理指令（宏定义)

```c
const int MAX_USERS = 100;       // 整型常量
const float PI = 3.14159f;       // 浮点型常量
const char NEWLINE = '\n';       // 字符型常量
```

Listing 5: const 关键字

```c
#define MAX_SIZE 50
#define TAX_RATE 0.15
#define DEBUG_MODE true
```

Listing 6: 宏定义