1. 结构体嵌套二级指针练习
2. 结构体偏移量
   1. 可以利用offsetof来计算结构体中属性的偏移
   2. 也可以通过地址的相加运算 计算偏移量
   3. 结构体嵌套结构体
3. 内存对齐
   1. 内存对齐原因：以空间换时间
   2. 对于自定义数据类型对齐规则：
      1. //1、从第一个属性开始 偏移为0
      2. //2、第二个属性开始，地址要放在 该类型整数倍 与 对齐模数比 取小的值 的整数倍上
      3. //3、所有的属性都计算结束后，整体再做二次对齐，整体需要放在属性中最大类型 与 对齐模数比 取小的值的整数倍上
   3. 如果查看对齐模数
      1. #pragma pack(show)
      2. 默认对齐模数 是8，可以将对齐模数改为 2的n次方
   4. 当结构体嵌套结构体时候，只需要看子结构体中最大数据类型就可以了
4. 文件读写回顾
   1. 按照字符进行读写
      1. 写文件 fputc
      2. 读文件 fgetc
      3. 文件结尾 EOF END OF FILE
   2. 按行读写
      1. 写文件 fputs
      2. 读文件 fgets
   3. 按块读写
      1. 写文件 fwrite
         1. 参数1 数据地址 参数2 块大小 参数3 块个数 参数4 文件指针
      2. 读文件 fread
   4. 按格式化读写
      1. 写文件fprintf
      2. 读文件fscanf
   5. 随机位置读写
      1. fseek （文件指针，偏移， 起始位置 SEEK\_SET SEEK\_END SEEK\_CUR）
      2. rewind （文件指针） 将文件光标置首
      3. error宏 全局变量 perror打印宏的提示错误信息
5. 文件读写注意事项
   1. 当按照字符的方式读文件时候，通常利用判断EOF获取是否读到文件尾
   2. 当对自定义数据类型写入文件时，不要将指针写入到文件里，要将指针指向的内容写入
6. 配置文件读写
   1. 需求：将文件中的有效内容截取出来，并且放入到一个键值对的数组中
      1. struct ConfigInfo { char key[64] ; char value[64] };
   2. 获取有效行数
   3. 判断当前行是否有效
   4. 解析数据 parseFile
      1. 将有效数据放入到数组中，数组在堆区开辟
   5. 根据key获取value getInfoByKey
   6. 释放内存 freeSpace