1. 宏函数
   1. #define MYADD(x,y) ((x) + (y)
   2. 将一些频繁短小的函数 写成宏函数
   3. 宏函数优点：以空间换时间
   4. 普通函数有入栈、出栈时间开销
2. 函数调用流程
   1. 局部变量、函数形参、函数返回地址.. 入栈 和 出栈
3. 调用惯例
   1. 主调函数和被调函数必须要有一致约定，才能正确的调用函数，这个约定我们称为调用惯例
   2. 调用惯例 包含内容： 出栈方、参数传递顺序、函数名称修饰
   3. C/C++下默认调用惯例： cdecl 从右到左 ，主调函数管理出栈
4. 栈的生长方向和内存存放方向
   1. 栈生长方向
      1. 栈底 --- 高地址
      2. 栈顶 --- 低地址
   2. 内存存放方向
      1. 高位字节数据 --- 高地址
      2. 低位字节数据 --- 低地址
      3. 小端对齐方式
5. 空指针和野指针
   1. 空指针
      1. 不能向NULL或者非法内存拷贝数据
   2. 野指针
      1. 指针变量未初始化
      2. 指针释放后未置空
      3. 指针操作超越变量作用域
   3. 空指针可以重复释放、野指针不可以重复释放
6. 指针的步长
   1. +1之后跳跃的字节数
   2. 解引用 解出的字节数
   3. 自定义结构体做步长练习
      1. 通过 offsetof( 结构体名称， 属性) 找到属性对应的偏移量
      2. offsetof 引入头文件 #include<stddef.h>
7. 指针的间接赋值
   1. 三大条件
      1. 一个普通变量+指针变量（ 实参+形参）
      2. 建立关系
      3. 通过\* 操作内存
   2. 利用Qt实现 操作地址 修改内存
8. 指针做函数参数的输入输出特性
   1. 输入特性
      1. 在主调函数中分配内存，被调函数使用
      2. 分配在栈上和堆区
   2. 输出特性
      1. 在被调函数中分配内存，主调函数使用
9. 字符串强化训练
   1. 字符串结束标志 \0
   2. sizeof 和 strlen
   3. 拷贝字符串 利用三种方式
      1. 利用[]
      2. 利用指针
      3. while (\*dest++ = \*src++){}
   4. 翻转字符串
      1. 利用[ ]
      2. 利用指针
10. sprintf使用
    1. 格式化字符串
    2. sprintf(目标字符串，格式化内容，占位参数…)
    3. 返回值 有效字符串长度