**巧用const**

1. Calloc（会有初始化默认值）
   1. 全局 const 保存在常量区（不可修改）
   2. 静态 const 保存在栈区，可以通过地址来进行修改
   3. 那我们的 const 一般会用来哪里
   4. 加入我们现在有一个结构体，占 80 字节，那我们现在需要打印出来
      1. PrintPerson(Person person); //这样是不是我们每次传进去的是一个结构体字节，要是我们循环100次，那是不是的8000.效率是不是很慢
      2. 那我们的解决方法是传进去地址，我们知道指针所占的字节是 4 个，要是我们循环100次，也才是400个
      3. PrintPerson(Person \*person);
      4. 但是地址传递，其修改值是很简单的，所以我们一般会用 const 来进行修饰一下
      5. 代码

typedef struct per{

char name[20];

int age;

char sex[8];

}Person;

void printPerson(const Person \*person){

printf("%s -- %d -- %s", person->name, person->age, person->sex);

}

void test01(){

printf("%d\n", sizeof(Person)); //打印一下所占的字节

Person person = {"weierLin", 23, "male"};

printPerson(&person);

}