1. block实现的原理/本质是什么？

block也是一个OC对象，转成C++代码的话，其实也是一个结构体。block帮我们封装了函数调用，以及函数调用所需要的OC对象。

2. block有哪些种类？

(1) globalBlock（内存中存放在数据段）

没有访问过 auto 变量的都是 globalBlock，访问了 static 的局部变量、全局变量都是 globalBlcok

(2) 栈block/stackBlcok（内存中存放在栈上）访问了 auto 变量的就是 stackBlcok（MRC环境），ARC环境下会帮我们自动调用copy方法拷贝到堆上

(3) 堆block/mallocBlock（内存中存放在堆上）

3. ARC哪些情况下会帮我们把 stackBlock 自动拷贝到堆上

（1）把 block 复制给一个强引用的对象时

（2）cocoa API使用的 UsingBlock

（3）gcd API 里面使用到的block

（4）block作为函数返回值时

4. block的属性为什么要用copy修饰？

block如果不使用 copy 进行修饰，那栈上的 block 就不会被拷贝到堆上，block代码出了 } 就自动销毁了，当再执行 block 里面代码时，就会出现野指针现象。

当然在ARC环境下，也可以用strong修饰，MRC环境一定要用copy

5. block内部可以修改外面的局部变量么？如何解决？

block 内部不能修改 auto 变量，因为 block 在访问局部变量时，只是把局部变量捕获到 block 中，转成C++代码的话，可以看到 block 结构体里面有一个变量

解决方案1，可以把 auto 变量修改为 static 变量（原理：捕获的是变量指针）

解决方案2，可以把 auto 变量修改为全局变量（原理：直接访问，不需要捕获）

解决方案3，可以使用\_\_block关键字（原理：block内部又会创建一个对象，这个对象里面有这个变量）

6. 使用block有哪些使用注意？

使用block要注意循环引用的问题。解决方案：可通过\_\_weak、\_\_unsafe\_unretained 关键字解决。当然也可以使用\_\_block关键字

7. \_\_block的作用是什么？有什么使用注意点？

（1）\_\_block可以修改外部变量。它会帮我们把变量封装成一个对象，也就是一个结构体；

（2）\_\_block可以解决循环引用的问题（必须执行这个block，block里面还需要让对象(person) = nil）。

使用\_\_block 需要注意下内存管理，而且在 MRC 环境中，\_\_block不会对OC对象进行强引用（没有retain 操作）；

\_\_block 不能修饰全局变量、static 变量

8. block在修改 NSMutableArray，需不需要添加 \_\_block?

如果需要修改 array 的值（array = nil）则是需要的，

如果只是使用 array （[arrary addObject:@""]）则是不需要的。