go指针 1、类型指针，允许对这个指针类型的数据进行修改，传递的数据可以直接使用指针，无须拷贝数据，类型指针不能偏移和运算2、切片，由指向起始元素的原始指针、元素数量和容量组成

ptr := &house //对字符串取地址，将指针保存到变量ptr中

fmt.print(“%T”,ptr) //res: \*string

value:=\*ptr //对ptr指针进行取值，变量value的类型变成string

指针交换

func swap(a,b \*int){

t:=\*a

\*a=\*b //取b指针的值，赋给a指针指向的变量

\*b=t

}

go 切片slice

由数组、len、cap，在append扩容时，会查看数组后面有没有连续内存块，没有就重新生成大的

go数组

数组是值类型，赋值和传参都会复制整个数组数据（地址变化）。数组长度不可变。

map不是线程安全的

go map通过hash表实现

**var map\_var map[key\_data\_type]value\_data\_type**

**map\_var := make(map[key\_data\_type]value\_data\_type)**

value, ok := map\_var[key]

**Go部分面试题**

1. **等待一百个协程执行完。var w syn.WaitGroup。w.add(1) go func{w.done}。w.wait()**
2. **可以限制线程数量(初始一万SetMaxThreads)。**
3. **可以从已经关闭的chan读数据，v,ok= <-ch读完返回默认值和false**
4. **goroutine数量受限制于内存大小。**
5. **goroutine协程池。（通过启动多少个goroutine的数量的channel）**
6. **单个协程的栈内存2KB**
7. **go defer是栈结构**
8. **类型的实例不可比较，指针类型。**
9. channel实现协程的同步，

for range ch{

count--

if 0==count{

close(ch)

}

}

1. map无顺序，通过slice让key有序。

sort.Strings(slice) //进入地址进行改变

11.