

1、看下面一段程序，回答问题

```
1 s1 := make([]int, 3, 4)
2 s2 := append(s1, 1)
```

请问s1、s2内各有什么元素？

s1修改一个元素会影响s2吗？s2修改一个元素会影响s1吗？

s2再增加一个元素会怎么样？

```
1 s1 := make([]int, 3, 4)
2 s2 := append(s1, 1)
3 fmt.Printf("%v %p %p %d %d\n", s1, &s1, &s1[0], len(s1), cap(s1))
4 fmt.Printf("%v %p %p %d %d\n", s2, &s2, &s2[0], len(s2), cap(s2))
5 // 如果s2再增加一个元素，就会分道扬镳
```

上面代码中，s1、s2共用底层数组，且s1、s2都从头使用底层数组，容量一样。所以，s1修改元素会影响s2，s2修改元素不一定影响s1。s2再增加一个元素会扩容。

```
1 s1 := make([]int, 3, 4)
2 s2 := append(s1, 1)
3 fmt.Printf("%v %p %p %d %d\n", s1, &s1, &s1[0], len(s1), cap(s1))
4 fmt.Printf("%v %p %p %d %d\n", s2, &s2, &s2[0], len(s2), cap(s2))
5 fmt.Println("~~~~~")
6 // 如果s2再增加一个元素，就会分道扬镳
7 s2 = append(s2, 2)
8 fmt.Printf("%v %p %p %d %d\n", s1, &s1, &s1[0], len(s1), cap(s1))
9 fmt.Printf("%v %p %p %d %d\n", s2, &s2, &s2[0], len(s2), cap(s2))
```

2、有一个数组 [1,4,9,16,2,5,10,15]，生成一个新切片，要求新切片元素是数组相邻2项的和。

```
1 s := [...]int{1, 4, 9, 16, 2, 5, 10, 15}
2 fmt.Println(s)
3 length := len(s) - 1
4 out := make([]int, 0, length)
5 for i := 0; i < length; i++ {
6     out = append(out, s[i]+s[i+1])
7 }
8 fmt.Println(out)
```

3、数字重复统计

- 随机产生100个整数
- 数字的范围[-100, 100]
- 降序输出这些生成的数字并打印其重复的次数

```
1 r.Intn(201) // 返回[0, 201)，在整数范围就是[0, 200]
2 r.Intn(201) - 100
```

```
1 package main
2
3 import (
4     "fmt"
5     "math/rand"
6     "sort"
7     "time"
8 )
9
10 func main() {
11     limiter := 201           // 界限
12     total := 100             // 生成总数
13     nums := make([]int, 0, total) // 放生成的数据的容器
14
15     r := rand.New(rand.NewSource(time.Hour.Nanoseconds()))
16
17     counter := make(map[int]int) // 统计数据存储容器
18     keys := make([]int, 0, total) // 最多100个不一样的key
19
20     for i := 0; i < total; i++ {
21         n := r.Intn(limiter) - 100 // [-100, 100]
22         nums = append(nums, n)     // 存起来
23         if _, ok := counter[n]; !ok {
24             counter[n] = 0         // 不存在就创建
25             keys = append(keys, n) // 一个全新的key
26         }
27         counter[n] += 1
28     }
29
30     fmt.Println(nums)
31     fmt.Println(counter, len(counter))
32     fmt.Println("~~~~~")
33
34     // sort, map不能直接使用sort的函数排序，map排序一般是对key排序
35     sort.Sort(sort.Reverse(sort.IntSlice(keys))) // 把去重后不同的int值降序排列
36     for _, k := range keys {
37         fmt.Println(k, counter[k])
38     }
39 }
```