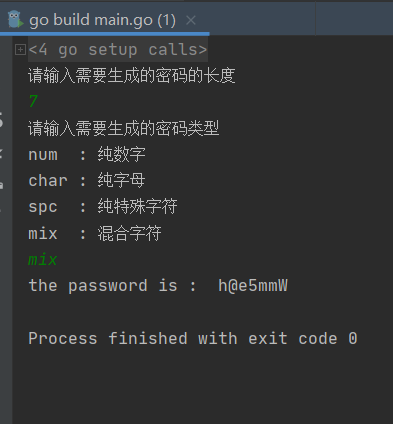
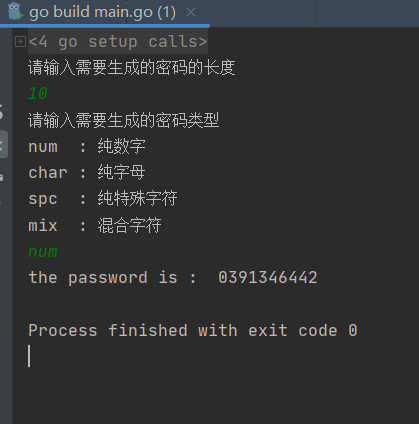
### 作业1（自拟题目）：生成密码，用户指定长度，指定密码类型，随机生成一串密码

运行截图



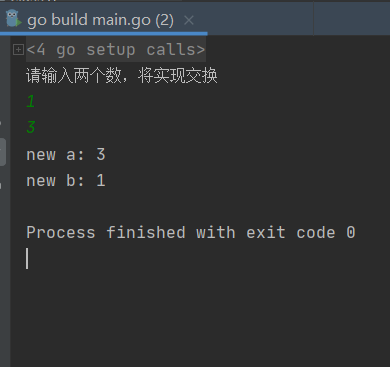


运行代码：

package main  
  
import (  
 "fmt"  
 "math/rand"  
 "time"  
)  
  
func main() {  
 var length int  
 var pType string  
 numstr:="0123456789"  
 charstr:="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"  
 spcstr:="`~!@#$%^&\*()-\_+=|?"  
 mixstr:=numstr+charstr+spcstr  
  
 fmt.Println("请输入需要生成的密码的长度")  
 fmt.Scanln(&length)  
 fmt.Println("请输入需要生成的密码类型")  
 fmt.Println("num : 纯数字")  
 fmt.Println("char : 纯字母")  
 fmt.Println("spc : 纯特殊字符")  
 fmt.Println("mix : 混合字符")  
 fmt.Scanln(&pType)  
  
 switch pType {  
 case "num":  
 password,ok:=getpassword(length,numstr)  
 if ok {  
 fmt.Println("the password is : ",password)  
 }  
 case "char":  
 password,ok:=getpassword(length,charstr)  
 if ok {  
 fmt.Println("the password is : ", password)  
 }  
 case "spc":  
 password,ok:=getpassword(length,spcstr)  
 if ok {  
 fmt.Println("the password is : ",password)  
 }  
 case "mix":  
 password,ok:=getpassword(length,mixstr)  
 if ok {  
 fmt.Println("the password is : ",password)  
 }  
 default:  
 fmt.Println("your input is not true!")  
 }  
  
}  
func getpassword(length int,str string ) (password string,ok bool) {  
 rand.Seed(time.Now().Unix())  
  
 for i:=0;i<length;i++{  
 j:=rand.Intn(len(str))  
 password=password+string(str[j])  
 }  
 ok=true  
 return password,ok  
}

# 作业2：编写一个交换函数供主函数使用，实现两个数的交换

运行截图：

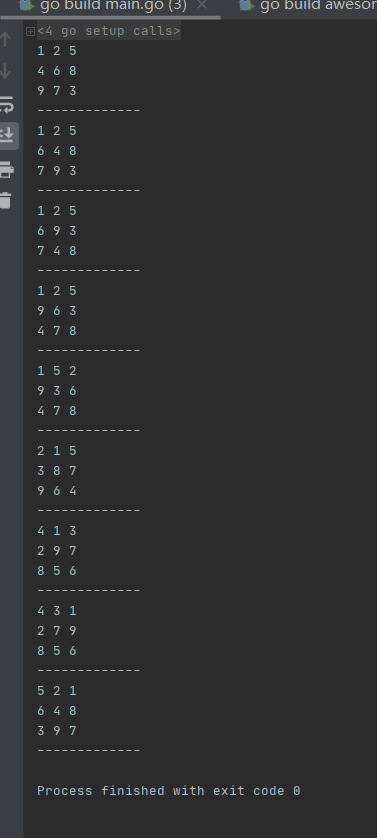


运行代码：

package main  
  
import "fmt"  
  
func main() {  
 fmt.Println("请输入两个数，将实现交换")  
 var a,b int  
 fmt.Scanln(&a)  
 fmt.Scanln(&b)  
 a,b=swap(a,b)  
 fmt.Println("new a:",a)  
 fmt.Println("new b:",b)  
  
}  
func swap(a,b int)(int ,int ) {  
 return b,a  
}

# 作业3：简单的数独游戏

运行截图：



运行代码：

package main  
  
import (  
 "fmt"  
 "strconv"  
)  
  
func main() {  
 //使用穷举法  
 var arr [10] int  
 for i := 111111111; i < 999999999; i++ {  
 arr=getarr(arr,i)  
 if test(arr){  
 show(arr)  
 fmt.Println("-------------")  
 }  
 }  
}  
func getarr(arr [10]int,num int) [10]int {  
 //将每个数都存放到数组的每个元素里  
 var numstr string  
 numstr=strconv.Itoa(int(num))//将数字转为字符串  
 for i:=1;i<=len(numstr);i++ {  
 arr[i]=int(numstr[i-1]-48)//将字符串中的每个数字转为int型放入数组  
 }  
 return arr  
}  
func test(arr [10]int) bool {  
 //将条件分开写，尽量提高运行效率  
 //条件1  
 if arr[1]+arr[2]+arr[3] == 8 &&  
 arr[4]+arr[5]+arr[6] == 18 &&  
 arr[7]+arr[8]+arr[9] == 19 &&  
 arr[1]+arr[4]+arr[7] == 14 &&  
 arr[2]+arr[5]+arr[8] == 15 &&  
 arr[3]+arr[6]+arr[9] == 16 {  
 } else {  
 return false  
 }  
 //条件2，每个数不能为0  
 for i:=1;i<=9;i++{  
 if arr[i]==0{  
 return false  
 }  
 }  
 //条件2，各个数不能相同  
 for i:=1;i<=9;i++{  
 for j:=i+1;j<=9;j++{  
 if arr[i]==arr[j] {  
 return false  
 }  
 }  
 }  
 return true  
}  
  
func show(arr [10]int) {  
 for i:=1;i<=9;i++{  
 fmt.Print(arr[i]," ")  
 if i%3==0 {  
 fmt.Println("")  
 }  
 }  
}