1、写出下面代码输出内容。

```
package main

import (
        "fmt"
)

func main() {
        defer_call()
}

func defer_call() {
        defer func() { fmt.Println("打印前") }()
        defer func() { fmt.Println("打印中") }()
        defer func() { fmt.Println("打印后") }()

        panic("触发异常")
}
```

2、 以下代码有什么问题，说明原因

```
type student struct {
        Name string
        Age  int
}

func pase_student() {
        m := make(map[string]*student)
        stus := []student{
                {Name: "zhou", Age: 24},
                {Name: "li", Age: 23},
                {Name: "wang", Age: 22},
        }
        for _, stu := range stus {
                m[stu.Name] = &stu
```

```
        }
}
```

3、下面的代码会输出什么，并说明原因

```
func main() {
        runtime.GOMAXPROCS(1)
        wg := sync.WaitGroup{}
        wg.Add(20)
        for i := 0; i < 10; i++ {
                go func() {
                        fmt.Println("i: ", i)
                        wg.Done()
                }()
        }
        for i := 0; i < 10; i++ {
                go func(i int) {
                        fmt.Println("i: ", i)
                        wg.Done()
                }(i)
        }
        wg.Wait()
}
```

4、下面代码会输出什么?

```
type People struct{}

func (p *People) ShowA() {
        fmt.Println("showA")
        p.ShowB()
}
func (p *People) ShowB() {
        fmt.Println("showB")
}

type Teacher struct {
        People
}
```

```
func (t *Teacher) ShowB() {
        fmt.Println("teacher showB")
}

func main() {
        t := Teacher{}
        t.ShowA()
}
```

5、下面代码会触发异常吗？请详细说明

```
func main() {
        runtime.GOMAXPROCS(1)
        int_chan := make(chan int, 1)
        string_chan := make(chan string, 1)
        int_chan <- 1
        string_chan <- "hello"
        select {
        case value := <-int_chan:
                fmt.Println(value)
        case value := <-string_chan:
                panic(value)
        }
}
```

6、下面代码输出什么?

```
func calc(index string, a, b int) int {
        ret := a + b
        fmt.Println(index, a, b, ret)
        return ret
}

func main() {
        a := 1
        b := 2
        defer calc("1", a, calc("10", a, b))
        a = 0
```

```
        defer calc("2", a, calc("20", a, b))
        b = 1
}
```

7、请写出以下输入内容

```
func main() {
        s := make([]int, 5)
        s = append(s, 1, 2, 3)
        fmt.Println(s)
}
```

8、下面的代码有什么问题?

```
type UserAges struct {
        ages map[string]int
        sync.Mutex
}

func (ua *UserAges) Add(name string, age int) {
        ua.Lock()
        defer ua.Unlock()
        ua.ages[name] = age
}

func (ua *UserAges) Get(name string) int {
        if age, ok := ua.ages[name]; ok {
                return age
        }
        return -1
}
```

9、下面的迭代会有什么问题?

```
func (set *threadSafeSet) Iter() <-chan interface{} {
        ch := make(chan interface{})
        go func() {
                set.RLock()
```

```
                for elem := range set.s {
                        ch <- elem
                }

                close(ch)
                set.RUnlock()

        }()
        return ch
}
```

10、以下代码能编译过去吗？为什么？

```
package main

import (
        "fmt"
)

type People interface {
        Speak(string) string
}

type Stduent struct{}

func (stu *Stduent) Speak(think string) (talk string) {
        if think == "bitch" {
                talk = "You are a good boy"
        } else {
                talk = "hi"
        }
        return
}

func main() {
        var peo People = Stduent{}
        think := "bitch"
```

```
        fmt.Println(peo.Speak(think))
}
```

11、以下代码打印出来什么内容，说出为什么。。。

```
package main

import (
        "fmt"
)

type People interface {
        Show()
}

type Student struct{}

func (stu *Student) Show() {

}

func live() People {
        var stu *Student
        return stu
}

func main() {
        if live() == nil {
                fmt.Println("AAAAAAA")
        } else {
                fmt.Println("BBBBBBB")
        }
```

链接：https://zhuanlan.zhihu.com/p/26972862