

6月10日综合练习：

(10分) 1、“Go is an open source programming language that makes it easy to build simple, reliable, and efficient software.” (From the Go web site at golang.org)

Go 语言是一门开源的程序设计语言，它可以使得构建简单、可靠、高效的软件变得容易。请对 Go 中的以下几项具体语言特征进行评估分析。

- 有丰富的内置类型，如 Slice 和 Map；
- 自动垃圾回收。
- 函数多返回值。
- 错误处理机制。

(15分) 2、In general, languages can be formally defined in two distinct ways: by recognition and by generation.

通常，语言的语法可以以两种不同的方式来形式化地描述：生成和识别。比如，正则表达式和上下文无关语法可用来生成一个语言的句子，都可以被看作语言生成器；状态转化图可以决定一个给定的句子是否属于一个语言，可被看成是语言识别器。

考虑这样一个正则表达式： $i(rb|fb)^*o$

- (1) 请给出与这个正则表达式描述相同语言的状态转化图；
- (2) 请用上下文无关语法描述这个语言。

(15 分) 3、考虑如下的上下文无关文法：

$$\begin{aligned} S &\rightarrow E \\ E &\rightarrow E + E \mid -E \mid id \end{aligned}$$

(1) 这个文法有没有歧义？请证明。

(2) 如果有歧义，请改写成等价的非歧义文法。在改写中，一元负“-”的优先级高于二元加法“+”的优先级，二元加法“+”为左结合。考虑到便于支持递归下降语法分析的实现，改写中请避免出现左递归的形式。

(10 分) 4、考虑下面的程序并回答问题：

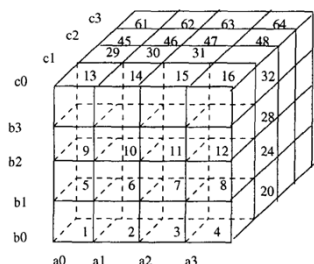
```
fun g()
{
    int a=10;
    void f(int b) {
        b = b*a;
        a = a-b;
        print(b);
    }
    {
        int a = 8;
        f(a);
        print(a);
    }
}
```

对于下面每一种情况，分别给出所有的输出结果，并简要解释原因：

- (1) 按值传递和静态作用域；
- (2) 按值传递和动态作用域；
- (3) 按引用传递和静态作用域；
- (4) 按引用传递和动态作用域。

(15 分) 5、(1) 假设有一个三维数组 $R[40][400][4000]$ ，三个维分别用 A, B, C 表示，最小下标都为 0。每个元素的大小为 1 个字节。如果使用行优先的方式实现数组的存储，元素 $R[5][6][7]$ 的存储地址相对于数组首元素地址的偏移是多少？列优先呢？给出计算公式和结果。

(2) 考虑一种新的数组实现策略。每个维分四等份，整个数组共分为 64 个三维小块进行存储。假设用求和的方式计算三个二维数组 AB, BC, AC 。如果要求顺序扫描一遍三维数组 R ，并行计算出这三个二维数组，在计算过程中，每计算出一个元素，就将它输入，不再占用内存。为了避免重复遍历小块，需要把部分计算出来的中间值保存下来。那么对于三维数组 R 是采用行优先还是列优先的方式来访问，计算所需的缓存较少？请说明理由。



$$AB[i][j] = \sum_{k=0}^{3999} R[i][j][k],$$

$$BC[j][k] = \sum_{i=0}^{39} R[i][j][k],$$

$$AC[i][k] = \sum_{j=0}^{399} R[i][j][k].$$

(15 分) 6. 考虑下面的程序，任选其一，请画出当执行到达位置 1 时，运行时栈中所有的活动记录实例，连接它们之间的静态链和动态链。

(1)

```
func main(){
    doc, err := html.Parse(os.Stdin)
    if err != nil {
        fmt.Fprintf(os.Stderr, "outline: %v\n", err)
        os.Exit(1)
    }
    outline(nil, doc)
}
// outline uses recursion over the HTML node tree to print the structure of the tree. Push and
// print the stack.
func outline(stack []string, n*html.Node) {
    if n.Type == html.ElementNode {
        stack = append(stack, n.data) //push tag
        fmt.Println(stack)
    }
    for c := n.FirstChild; c!=nil; c=c.NextSibling {
        outline(stack, c)
    }
}
}
./fetch https://golang.org | ./outline
[html]
[html head]
[html head meta]
[html head title]      ←1
[html head link]
[html body]
[html body div]
[html body div div]
[html body div form]
```

(2)

```
func main(){
    type tree struct {
        value int
        left, right *tree
    }
    func Sort(values []int) {
        var root *tree
        for _, v := range values {
            root = add(root, v)
        }
        appendValues(values[:0], root)
    }
    func add(t *tree, value int) *tree {
        if t == nil {
            t = new(tree)
            t.value = value
            return t
        }
        if value < t.value {
            t.left = add(t.left, value)      ←1
        } else {
            t.right = add(t.right, value)
        }
        return t
    }
    func appendValues(values []int, t *tree) []int {
        if t != nil {
            values = appendValues(values, t.left)
            values = append(values, t.value)
            values = appendValues(values, t.right)
        }
        return values      ←2
    }
    list := []int{8,3,9}
    Sort(list)
}
```

（20 分）7. 假设由你来设计一个适用于智能领域（或你感兴趣的特定领域）的语言，你会进行哪些设计？

以下为一些提示（可以包含但不限于这几方面）：

你怎么理解“所有能够由流程图表述的算法都可以仅用二分支的选择语句和逻辑控制的循环语句在程序设计语言中编码”？如果设计变量，如何考虑变量的类型检测问题？变量的存储管理问题？