

1992 年浙江大学计算机科学基础（包括程序设计）考研试题

考研加油站收集整理 <http://www.kaoyan.com>

一. 完成下列各题 (每题 6 分)

1. 已知 $a = \text{TRUE}$, $b = \text{FALSE}$, $x = 6$, $i = 4$, $j = 5$, $t = 3$, $u = 2$
求出 PASCAL 表达式的值:

$a \text{ AND NOT } (x > 5) \text{ AND } (i > j + 5) \text{ OR } a > b \text{ AND } b$

2. 已知计算闰年的法则如下:

被 400 除尽的年份是闰年, 或者

被 4 除尽又不被 100 除尽的年份也是闰年。

试写出判断闰年的 P 语言。

3. 试设计一记录型数据结构, 用来表达某一块区下列地理特征的信息:

a. 区域名称, b. 区域大小 (面积), c. 区域中心坐标 (经度, 纬度), d. 区域的边界 (用边界上一定间隔的点的坐标序列表示)。

二. 试编写一个 PASCAL 程序, 它能够对输入的 6 个整数按升序排列并打印出相应的次序。例如 输入

10 33 4

7, 4, 8, 9, 1, 5 则打印

输出为:

(10分)

7	4	8	9	15
5	7	4	8	9
1	5	7	4	8
9	1	5	7	4
8	9	1	5	7
4	8	9	1	5

三. 已知一程序为:

```

PROCEDURE unknown(a, b: integer);
BEGIN
  writeln('a=', a, 'b=', b);
  IF (a MOD b) = 0
  THEN write('The result is', b)
  ELSE unknown(b, a MOD b)
END

```

试写出执行 unknown(234, 63) 时所有打印的结果。(10分)

四. 著名的罗马史学家 Josephus 提出一个称为 Josephus 问题可用下列例子来说明: m 只猴子要选大王, 选举办法如下: 所有的猴子按 $1 \sim m$ 编号围坐一圈, 从第一号开始按顺序 $1, 2, 3, \dots, n$ 报数, 凡报到 n 号的退出圈外, 如此循环报数, 直到圈内只剩下一只猴子时, 这只猴子就是大王。

1. 试设计该问题求解的算法或框图。
2. 写出对应的 PASCAL (或 C 语言) 程序 (m, n 由用户输入, 猴王号码打印出来)。(12分)

五. 说出下列 PASCAL 函数的功能, 计算 $A(6)$ 的值, 并写出非递归形式的 PASCAL 程序。 (10分)

```
function A(n: nonegint): nonegint;
var   x, y: nonegint;
begin
    if n <= 1
    then A := n
    else begin
            x := A(n-1);
            y := A(n-2);
            A := x + y;
        end
end;
```

六. 一种用于表达二元表达式(例: $3 + 4 * (5 - 7) / 5 + 3$)的二叉树称为异类二叉树, 它的叶结点包含数字, 非叶结点包含代表操作符的字符, 下面是使用变体记录说明的该结点的信息部分:

```
type infotype = (operand, operator);
    nodeptr = ↑ nodetype;
    nodetype = record
        left: nodeptr;
        right: nodeptr;
        case tag: infotype of
            operator: (chinfo: char);
            operand: (numinfo: real);
        end;
```

试编写一函数,它的自变量为指向该异类二叉树的指针,返回由该树所表达的表达式式的值。(15分)

七. 简述 Huffman 算法, 并证明它的确能生成一颗带权外部路径最小的树。(10分)

八. 算法 delete 从 Hash 表中删去一项, 试分析其执行过程, 并举例说明。(15分)

```
delete(i)
```

```
    int j, r
```

```
    while(1) {
```

```
        table[i] = 0
```

```
        j = i
```

```
        while(1)
```

```
            i = (i - 1 + M) mod M
```

```
            if (table[i] == 0)
```

```
                return
```

```
            r = hash(table[i].key)
```

```
            if (!(i ≤ r < j || r < j < i || j < i ≤ r))
```

```
                break
```

```
            table[j] = table[i]
```

```
    }
```