

## 第 7 次（期中）上机解题报告

15081070 张雨任

### 一、A 题

1. 题目代号与评测记录序号：{A277613}

2. 解题思路：

a 和 b 虽然都在 int 范围内，但是 a+b 可能达到了 long，所以 a 和 b 都要声明为 long，否则 a+b 还是 int。

### 二、B 题

1. 题目代号与评测记录序号：{B279566}

2. 解题思路：

这道题考察链表的操作，题很长，输入输出很繁杂，所以容易出错。这里就着重讲一下自己出错的地方吧，好好反省一下

①（delete 操作）删除链表的时候带着头结点也删除了。

②（delete 操作）删除结点的时候删成了要删除的结点前面的结点了。

③（erase 操作）erase 操作上机的时候写的很乱，错误很多，下来之后重新写了一个。这里说一下思路：erase x 是删除和 x 值相同的结点，并且返回删除成功与否以及删除成功的话被删除的结点的位置。首先，如果是空表，则返回 0，删除失败。然后设置 found 值，设置指针 p 来遍历链表，如果有和 x 相等的值，found 置为 1，并跳出。否则，就向后移动指针，同时 i（代表 x 的位置的计数器）自增。之后，如果 found 为 1，说明找到了，那么遍历链表直到 p 的前一个，设为 r，再把 p 的下一个在设为 s，就出现了 r, p, s 三个连续的指针，其中 p 是匹配了 x 的指针，删除它并让 r 的 next 指向 s 即可。如果 found 为 0，说明没找到，返回 0 即可。

这道题很复杂，但是并不难，没做出来很不应该。上机还是要保持心态稳定。

### 三、C 题

1. 题目代号与评测记录序号：{C278187}

2. 解题思路：

用栈可以完全模拟进制转换中取余再倒置输出的方法，就是每次，把取余的结果放在栈中，商作为新的被除数，直到商等于 0 为止。之后把栈里的数依次弹出即可，弹出就是倒置输出的过程。

### 四、D 题

1. 题目代号与测评记录序号：{D278731}

2. 解题思路：

用头结点不在循环中的循环链表模拟约瑟夫环的整个过程。首先，这道题是要求从第 k 个人开始报数，所以要把起始结点移到第 k 个结点，从它开始每到第 m 个人就执行链表的删除。特殊考虑 m=1 的时候，m=1 的时候没有进

入 for 循环, 所以 pre 没有指向最后一个节点, 所以语句 `pre->next=p->next` 是非法的。所以当 m 等于 1 的时候, 把 pre 指向最后一个结点, 然后删除元首结点, 再执行 `pre->next=p->next` 即可。

这里再提供两种思路:

- ① 当 m 等于 1 的时候, 就是从元首结点依次向下删的过程, 所以当  $m=1$  的时候不进行 `pre->next=p->next` 的操作即可。
- ② 这种完全模拟第 m 个结点删除的方法比较耗时, 每次其实删除的就是第  $\text{tmp}=m\%(\text{当前人数})$  个结点, 特殊考虑  $\text{tmp}=1$  和  $\text{tmp}=0$  的时候, 等于 0 其实就是  $m=\text{当前人数}$ , 所以要遍历到最后一个结点, 等于 1 就是删除下一个结点, 所以先遍历到被删结点的前一个即可。这样比较省时间。

## 五、E 题

1. 题目代号与测评记录序号: {E278440}

2. 解题思路:

用队列来实现杨辉三角形。无论 n 是几都要先输出第一行的 1, 所以先输出之, 然后把第二行的两个 1 放进队列。每一行开始的时候都要把 0 进队。处理某一行时, 先执行退队, 把上一行的退出来为 t, 在把 s (也就是上一个循环的 t) 与 t 相加, 就可以得到他们下面的一个数, 之后输出此轮退队的 t, 注意每行末尾是 0, 所以不输出。循环 n-1 次, 代表再输出 n-1 行, 每行结束输出换行即可。

## 六、F 题

1. 题目代号与测评记录序号: {F279881}

2. 解题思路:

这道题采用递推的思路, 通过某一个数的值, 可以得到下一个数的值, 由于整个量级很大, 因此采用 string 来存储数据。第一个是 1, 下一个是 11, 因为 1 可以写成 “1 个 1”; 11 的下一个是 21, 因为 11 可以表示成 “2 个 1”; 21 的下一个可以写成 1211, 因为 21 可以写成 “1 个 2, 1 个 1” ……因此对于某数, 按位遍历, 先输出连续且相同的值的个数, 再输出该值, 比如对于 1211: 先输出连续的个数, 是 1, 然后是值的大小, 是 1; 再输出连续的个数, 是 1, 然后是值的大小, 是 2; 再输出连续的个数, 是 2, 然后是值的大小, 是 1。这样, 1211 之后的一个就是 111221。我设立了一个 char 类型的 flag 变量用来记录当前判断某串 (设为  $a[i-1]$ ) 中的某位数字是否和 flag 相同, 遍历字符串的每一位, 相同, 则累加计数器, 不同, 则要把计数器变为 string 添加到  $a[i]$  的尾部, 再把 flag (当前数字的值) 添加到  $a[i]$  的尾部, 之后把 cnt 置为 0, 把 flag 置为  $a[i-1]$  当前未能判断相等的数字。这里涉及到一个问题, 一个 int 类型的 cnt 变量, 要把它转化成 string 类型, 再放到  $a[i]$  的尾部, 下面讲一讲 int 转 string 的方法:

设 iData 为要变为 string 的 int 类型变量, sData 为从 int 转变过来的 string 类型变量, 那么转化方式为:

```
#include <sstream>
string sData;
int iData;
stringstream stream;
stream<<iData;
sData=stream.str();
这样就可以解决 int 转 string 了。
```

## 七、G 题

1. 题目代号与测评记录序号：{G279800}

2. 解题思路：

这道题需要使用递归的思路。

字符串 a 和 b 等价的定义：

定义一：若单个字符串长度为奇数，则两字符串只有完全相等才可称为等价。

定义二：若单个字符串长度为偶数，则将 a 从中间分为长度相等的两半，记为 a1 和 a2，将 b 从中间分为长度相等的两半，记为 b1 和 b2，满足 a1 与 b1 等价且 a2 与 b2 等价，或者 a1 与 b2 等价且 a2 与 b1 等价，称 a 与 b 等价。

解题思路如下：

输入两个长度相同的串。如果长度为奇数，则相同输出 YES，不同输出 NO。

如果长度为偶数，进入 cmp 函数，cmp 返回 1，代表等价，输出 YES，返回 0，代表不等价，输出 0。

下面讲讲递归函数 cmp：

cmp 函数输入两个参数，也就是用户输入的 a 和 b。对于进入 cmp 的 a 和 b，他们的长度一定是偶数。对于 a 和 b，把它们都分为长度相同的前后两部分，a 分为 a1 和 a2，b 分为 b1 和 b2。规则是“满足 a1 与 b1 等价且 a2 与 b2 等价，或者 a1 与 b2 等价且 a2 与 b1 等价，称 a 与 b 等价”。那么如果 a1a2b1b2 的长度是奇数，说明运用定义一来判断等价就可以，即：满足 a1==b1 且 a2==b2 说明等价，满足 a1==b2 且 a2==b1 说明等价。如果 a1a2b1b2 的长度是偶数，则要满足“a1 与 b1 等价且 a2 与 b 可以写成 2 等价，或者 a1 与 b2 等价且 a2 与 b1 等价”，就可以运用递归的思想：两个长度为偶数的串 x, y 等价可以写成 cmp(x, y)==1，那么可以写成：

```
if(cmp(a1,b1) && cmp(a2,b2)) return 1;
if(cmp(a1,b2) && cmp(a2,b1)) return 1;
else return 0;
```