Day 4A

报数

经典的约瑟夫环问题，最简单的做法就是用循环链表来模拟，直接每m个人将节点删除即可。
追求效率可以采用数组来链表，此处略。

敲7

与上一题类似，额外加入了初始循环条件，同时改变了出队条件。

以上两道题都可以用如下的流程模拟：

```
int cnt = 0;
while(cnt < n - 1)
{
    while(!check(t++))
        p = p->next;

    remove(&q);
    ++cnt;
}
```

括号匹配

这道题是一个简单的匹配问题，对于括号是否匹配，我们始终记录当前左括号的个数与右括号的个数差值即可，当差值<0或者最终结束时差值不为0即代表右括号不够。

对于题目要求的输出位置，我们可以采取在遇到左括号时将当前位置入栈，右括号时出栈并放至答案数组即可。

网页跳转

这道题目是一个简单的模拟，如图演示操作即可：

			WEB4	top, p
WEB3	top, p	VISIT WEB4	WEB3	
WEB2			WEB2	
WEB1	0		WEB1	0
WEB4	top, p		WEB4	top
WEB3		BACK	WEB3	p
WEB2			WEB2	
WEB1	0		WEB1	0
WEB4	top, p		WEB4	top
WEB3		BACK * 2	WEB3	
WEB2			WEB2	
WEB1	0		WEB1	0, p
WEB4	top, p		WEB4	top
WEB3		FORWARD	WEB3	
WEB2			WEB2	p
WEB1	0		WEB1	0
WEB4	top, p			
WEB3		VISIT WEB5	WEB5	top, p
WEB2			WEB2	
WEB1	0		WEB1	0

Day 4B

打印锯齿矩阵

直接开数组`int[10000][10000]`的大小$4 * 10^8 bytes$大于题目给定内存$1.31072 * 10^8 bytes$，故采用链表存储。

直接建立`n`个链表即可，最后每行输出即可。

求链表环的长度

首先令指针`f, b = head`，`f`每次推进两个单元，`b`每次推进一个单元，如果最终二者能够相遇则代表有环，再让`b`遍历一遍直至再次遇到`f`即可统计出长度。

注意下面程序中对`f->next`及`f->next->next`的意义。

找出两个链表的交点

首先统计出两个链表的长度，将较长者向后推进两者长度差，让二者处于统一开始位置，此时同步推进两个链表，若某一时刻相交则这个点必为交点。

1. 计算两个链表长度`len1`，`len2`
2. 将较长的链表向前推进`|len1 - len2|`
3. 两个指针同时向前推进