

概率题

某个地方天气有如下规律：如果第一天和第二天都不下雨，则第三天下雨的概率为30%；如果第一天和第二天中有任意一天下雨，则第三天下雨的概率为60%。问如果周一周二都没下雨，那么周四下雨的概率为_。

解题思路：首先将周一、周二、周三看成一组，求出周三下雨的概率，之后将周二、周三、周四看成一组从而求出周四下雨概率。

首先题目说，周一、周二都没下雨，由此可知，周三下雨的概率为30%，不下雨的概率为70%。所以周四下雨的概率为： $30\% * 60\% + 70\% * 30\% = 39\%$ 。

不用临时变量怎么实现swap(a,b)

- 加法实现：
 - $a = a + b; b = a - b; a = a - b;$
- 异或实现：
 - $a = a \oplus b; b = a \oplus b; a = a \oplus b;$
 - 原理： $a \oplus a = 0, a \oplus 0 = a.$

奇数个节点的单向链表，查找其中间节点（百度面试）

- 解题思路：设两个指针p，q。q每走一步，p走两步。当p到链表尾部的时候，q为中点位置。
- 关键代码：

```
void Find(LinkList &L)
{
```

```

    LNode *p, *q;
    p = q = L;
    while (p -> next) {
        p = p -> next -> next;
        q = q -> next;
    }
    cout << q -> data;
}

```

- 完整代码：[奇数个节点的单向链表，查找其中间节点](#)

判断一个单向链表是否形成了环形结构

- 设置两个指针同时从头结点出发，一个指针一次走一步，一个指针一次走两步。如果走的快的指针追上了走的慢的指针，就表明存在环形结构。

- 关键代码：

```

int FindLinkListCircle(LinkList &L)
{
    LNode *p = L;
    LNode *q = L;
    while (q && p && (p->next != NULL)) {
        p = p -> next -> next;

        if (p == q) {
            return 1; //表示存在环
        }
        q = q -> next;
    }
    return 0; //表示不存在环
}

```

- 完整代码：[判断一个单向链表是否形成了环形结构](#)