保龄球

【故事背景】

JYY 很喜欢打保龄球，虽然技术不高，但是还是总想着的高分。这里 JYY 将向你介绍他所参加的特殊保龄球比赛的规则，然后请你帮他得到尽量多的分数。

【问题描述】

一场保龄球比赛一共有 N 个轮次，每一轮都会有 10 个木瓶放置在木板道的另一端。每一轮中，选手都有两次投球的机会来尝试击倒全部的 10 个木瓶。对于每一次投球机会，选手投球的得分等于这一次投球所击倒的木瓶数量。选手每一轮的得分是他两次机会击倒全部木瓶的数量。

对于每一个**轮次**，有如下三种情况：

1、“**全中**”：如果选手第一次尝试就击倒了全部 10 个木瓶，那么这一轮就称为“全中”。在一个“全中”轮中，由于所有木瓶在第一次尝试中都被击倒，所以选手不需要再进行第二次投球尝试。同时，在计算总分时，选手在下一轮的得分将会被乘 2 计入总分。

2、“**补中**”：如果选手使用两次尝试击倒了 10 个木瓶，那么这一轮就称为“补中”。同时，在计算总分时，选手在下一轮中的第一次尝试的得分将会被乘以 2 计入总分。

3、“**失误**”：如果选手未能通过两次尝试击倒全部的木瓶，那么这一轮就被称为“失误”。同时，在计算总分时，选手在下一轮的得分会被计入总分，没有分数被翻倍。

此外，如果第 N 轮是“全中”，那么选手可以进行一次附加轮：也就是，如果第 N 轮是“全中”，那么选手将一共进行 $N+1$ 轮比赛。显然，在这种情况下，第 $N+1$ 轮的分数一定会被加倍。

附加轮的规则只执行一次。也就是说，即使第 $N+1$ 轮选手又打出了“全中”，也不会进行第 $N+2$ 轮比赛。因而，附加轮的成绩不会使得其他轮的分数翻番。

最后，选手的总得分就是附加轮规则执行过，并且分数按上述规则加倍后的每一轮分数之和。

JYY 刚刚进行了一场 N 个轮次的保龄球比赛，但是，JYY 非常不满意他的得分。

JYY 想出了一个办法：他可以把记分表上，他所打出的所有**轮次**的顺序重新排列，这样重新排列之后，由于翻倍规则的存在，JYY 就可以得到更高的分数了！

当然了，JYY 不希望做的太假，他希望保证重新排列之后，所需要进行的轮数和重排前所进行的轮数是一致的：比如如果重排前 JYY 在第 N 轮打出了“全中”，那么重排之后，第 N 轮还得是“全中”以保证比赛一共进行 $N+1$ 轮；同样的，如果 JYY 第 N 轮没有打出“全中”，那么重排过后第 N 轮也不能是全中。

请你帮助 JYY 计算一下，他可以得到的最高的分数。

【输入格式】

从文件 *bowling.in* 中读入数据。

第一行包含一个整数 N ，表示保龄球比赛所需要进行的轮数。

接下来包含 N 或者 $N+1$ 行，第 i 行包含两个非负整数 x_i 和 y_i ，表示 JYY 在这一轮两次投球尝试所得到的分数， x_i 表示第一次尝试， y_i 表示第二次尝试。

我们用 $x_i = 10, y_i = 0$ 表示一个“全中”轮。输入数据保证 $x_i + y_i \leq 10$ 。

读入数据存在 $N+1$ 行，当且仅当 $x_N = 10$ 且 $y_N = 0$ 。

【输出格式】

输出到文件 *bowling.out* 中。

输出一行一个整数，表示 JYY 最大可能得到的分数。

【样例输入】

```
2
5 2
10 0
3 7
```

【样例输出】

```
44
```

【样例说明】

按照输入顺序，JYY 将得到 37 分。

最佳方案是将 3 个轮次排列成如下顺序：

```
3 7
10 0
5 2
```

【数据规模与约定】

对于 20% 的数据满足 $N \leq 9$ ；

对于 50% 的数据满足 $N \leq 20$ ；

对于 100% 的数据满足 $N \leq 50$ 。