**卡牌进化**

**时间限制: 2000/1000 MS (Java/Others)**

**内存限制: 65536/65536 K (Java/Others)**

**问题描述**

小C最近沉迷某个卡牌游戏，他的卡组中只有两种牌：“血怒”和“祝福”。使用“祝福”后，小C会获得攻击力加1的增益，使用“血怒”后，小C的攻击力将变成原来的一半（向下取整），但是此后每次使用“祝福”都会额外获得1点的攻击力增益，且该效果可叠加，例如，如果小C在先前一共使用了两张“血怒”，那么接下来他使用“祝福”将会获得3点的攻击力增益。小C的初始攻击力为0。小C手里只能有一张牌，他可以选择使用或者丢弃手里的牌，然后获取下一张牌。现在小C预先知道了牌序，他想知道自己最后能获得的最大攻击力是多少。

**输入描述**

第一行一个整数n，表示一共有n张牌；

接下来一行n个整数,其中， 表示第i张牌是“祝福”，表示第i张牌是“血怒”

**输出描述**

一个整数，表示小C可以获得的最大攻击力

**输入样例1**

8

0 0 1 0 0 1 0 0

**输出样例1**

9

**样例解释1**

最佳的方案是除了第六张牌（血怒）之外均使用。

使用完前两张牌（祝福）后攻击力为2；

使用完第三张牌（血怒）后攻击力为1，攻击力增益变为2；

此后的几张牌只使用祝福而不使用血怒，共有4张祝福，因此最终攻击力为9。

**数据范围和说明**

30%的数据保证：

80%的数据保证：

100%的数据保证 ：