

病毒感染

【故事背景】

JOSI 的边陲小镇爆发了严重的 Jebola 病毒疫情，大批群众感染生命垂危。计算机科学家 JYY 采用最新的算法紧急研制出了 Jebola 疫苗，并火速前往灾区救治患者。

【问题描述】

一共有 N 个小镇爆发了 Jebola 疫情。这些小镇由于地处边陲，仅仅通过一条长直公路连接。方便起见我们将这些小镇按照公路连接顺序有 1 编号到 N 。JYY 会在第一天一早抵达 1 号小镇。

一开始在 i 号小镇，有 a_i 名患者感染了 Jebola 病毒。

每一天 JYY 可以选择：

1. 花费一天时间彻底治愈 JYY 目前所在的村庄的所有 Jebola 患者。这一天不会有任何患者死去。
2. 花费一天的时间前往一个相邻的村庄。

当一天开始时，如果一个村庄里有 k 个 Jebola 患者，那么这一天结束时，这 k 个患者会感染另外 k 个这个村子里的健康村民并死去。所以对于 i 号村庄，只要这个村庄没有被 JYY 彻底消灭疫情，那么每一天都会有 a_i 个村民死去。

JYY 希望采用措施使得疫情被整体消灭时，总共死去的村名数量尽量少。

为了达成这一目标，JYY 有时会选择抵达一个村庄但是并不对村民进行施救。这样的行为如果不加限制，往往会造成更加严重的后果。

试想这样的情形：假设当 JYY 第一次抵达村庄 i ，未作救治并直接前往了另一个村庄。那么由于 i 村庄的人们求生心切，一旦当 JYY 朝向靠近 i 村庄的方向前行时， i 村庄的村民就会以为 JYY 是来救他们了，而产生巨大的期望。之后倘若 JYY 再次掉头朝着远离 i 村庄的方向行进，那么 i 村庄的村民就会因为巨大的失落而产生绝望的情绪。

为了避免这种情况，JYY 对他的行程做了如下规定：

假设 JYY 进入 i 村庄并在第二天立即离开（村庄 i 的疫情并未治愈）。如果在之后的某一天，JYY 从村庄 j 前往村庄 k ，并满足 $|k - i| < |k - j|$ 。那么在之后的日子里 JYY 只能朝着 i 村庄前进直到抵达 i 村庄并立即治愈该村的患者。在前往 i 村庄的过程中，JYY 可以选择将途经村庄的疫情治愈。

比如，如果 JYY 有如下行程：

第一天：从村庄 1 前往村庄 2；第二天：从村庄 2 前往村庄 3；第三天：治愈村庄 3；第四天：前往村庄 2。

此时 JYY 对于之后三天的行程只有唯一一种选择：

第五天：治愈村庄 2；第六天：前往村庄 1；第七天：治愈村庄 1。

JYY 想知道在治愈所有村庄之前，至少会有多少村民因 Jebola 死去。

【输入格式】

从文件 *virus.in* 中读入数据。

输入第一行包含一个正整数 N 。

接下来一行包含 N 个整数，分别为 a_1, a_2, \dots, a_N 。

【输出格式】

输出到文件 *virus.out* 中。

输出一行一个整数，表示最优行程安排下会死去的村民数量。

【样例输入】

```
6
40 200 1 300 2 10
```

【样例输出】

```
1950
```

【样例解释】

我们用 $C(k)$ 表示治愈 k 号村庄， $i \rightarrow j$ 表示从村庄 i 前进到村庄 j ，用分号分隔每一天的行程安排，那么样例中的最优策略为：

$1 \rightarrow 2; C(2); 2 \rightarrow 3; 3 \rightarrow 4; C(4); 4 \rightarrow 3; C(3); 3 \rightarrow 2; 2 \rightarrow 1; C(1); 1 \rightarrow 2; 2 \rightarrow 3; 3 \rightarrow 4; 4 \rightarrow 5; 5 \rightarrow 6; C(6); 6 \rightarrow 5; C(5)$ 。

整个过程耗时 18 天。

【数据规模与约定】

对于 10% 的数据满足 $N \leq 10$;

对于 30% 的数据满足 $N \leq 20$;

对于 50% 的数据满足 $N \leq 60$;

对于 100% 的数据满足 $1 \leq N \leq 3000, 1 \leq a_i \leq 10^9$ 。