

预览输出

题目名称	委托
题目类型	传统型
可执行文件名	morning.exe
输入文件名	morning.in
输出文件名	morning.out
每个测试点时限	1.0 秒
内存限制	512 MiB
测试点数目	20
测试点是否等分	是

提交源程序文件名

对于 C++ 语言	morning.cpp
-----------	-------------

编译选项

对于 C++ 语言	<code>-O2 -std=c++14 -Wl,--stack=536870912</code>
-----------	-------------------------------------------------------

委托 (morning)

【题目背景】

从烦琐的跑腿，勘察，到雇凶杀人。只要付得起钱，他们就会接下任何工作。

他们受托于“事务所”，“辛迪加”，甚至是“翼”。从微乎其微的琐事，到惨绝人寰的要事。他们，无所不能。

【题目描述】

作为一名强大的收尾人，正如前文所说，你接下了来自 L 公司的 n 项委托。

每个委托的难度是不一样的，不过，你很强大，完全不用担心完成不了委托。只是不同的委托，因为难度不同，完成的时间也是不同的，对于委托 i 来说，你需要花费 A_i 的时间来完成。

这些委托之间互相都存在相互的联系，当你完成了其中任意一个委托时，其他所有的委托都因此变得更加简单。具体来说，当你完成了第 i 件委托时，会使得其他所有委托的完成需要的时间减少 i （当然，完成时间不可能变成负数，如果有的委托的完成需要的时间变成小于等于 0 时，这个委托你也立刻完成）。

你只有完成了一个委托，才可以去完成其他委托。

你听说 L 公司最近出现了一些奇怪的东西，为了安全起见，你需要尽快完成这些任务，然后赶紧离开。

那么，你至少需要多少时间才能完成全部的任务呢？

【输入格式】

从文件 *morning.in* 中读入数据。

第一行输入一个正整数 n

接下来一行一共 n 个正整数，第 i 个为 A_i

【输出格式】

输出到文件 *morning.out* 中。

输出最少的完成时间。

【样例 1 输入】

```
1 5
2 6 6 6 5 5
```

【样例 1 输出】

1 18

按照顺序完成委托即可

每个任务需要的时间为： $6 + 6 - 1 + 6 - 2 + 5 - 3 + 5 - 4 = 18$

【样例 1 解释】**【样例 2】**

见选手目录下的 *morning/morning2.in* 与 *morning/morning2.ans*。

该样例满足特殊性质

【样例 2 解释】**【样例 3】**

见选手目录下的 *morning/morning3.in* 与 *morning/morning3.ans*。

该样例满足 100% 的数据

【子任务】

对于 50% 的数据， $N \leq 10$

另有 20% 的数据，对于 $i \leq n$ ， $A_i \leq A_{i+1}$

对于 100% 的数据， $N \leq 10^5$ ， $A_i \leq 10^6$