

# 5. Хорошая строка

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

На день рождения маленький Ипполит получил долгожданный подарок — набор дощечек с написанными на них буквами латинского алфавита. Теперь-то ему будет чем заняться долгими вечерами, тем более что мама обещала подарить ему в следующем году последовательность целых неотрицательных чисел, если он хорошо освоит этот набор. Ради такого богатства Ипполит готов на многое. Прямо сейчас юный исследователь полностью поглощён изучением хорошеи строк. Хорошестью строки называется количество позиций от 1 до  $L - 1$  (где  $L$  — длина строки), таких, что следующая буква в строке является следующей по алфавиту. Например, хорошесть строки "abcdefghijklmnopqrstuvwxyz" равна 25, а строки "abdc" — только 1.

Ипполит размышляет над решением закономерной возникающей задачи: чему равна максимально возможная хорошесть строки, которую можно собрать, используя дощечки из данного набора? Вы-то и поможете ему с ней справиться.

## Формат ввода

Первая строка ввода содержит единственное целое число  $N$  — количество различных букв в наборе ( $1 \leq N \leq 26$ ). Обратите внимание: в наборе всегда используются  $N$  первых букв латинского алфавита.

Следующие  $N$  строк содержат целые положительные числа  $c_i$  — количество букв соответствующего типа ( $1 \leq c_i \leq 10^9$ ). Таким образом, первое число означает количество букв "a", второе число задаёт количество букв "b" и так далее.

## Формат вывода

Выведите единственное целое число — максимально возможную хорошесть строки, которую можно собрать из имеющихся дощечек.

### Пример 1

<b>Ввод</b> <input type="text"/>	<b>Вывод</b> <input type="text"/>
3	2
1	
1	
1	

### Пример 2

<b>Ввод</b> <input type="text"/>	<b>Вывод</b> <input type="text"/>
2	3
3	
4	

## Примечания

В первом тесте имеется по одной дощечке с каждой из 3 различных букв. Ответ 2 достигается на строке "abc"

Язык

```
1 #include <iostream>
2
3 int main() {
4     int N;
5     std::cin >> N;
6     unsigned long nice = 0;
7
8     int prev;
```

```
9      std::cin >> prev;
10     for (int i = 1, tmp; i != N; ++i) {
11         std::cin >> tmp;
12         if (tmp >= prev) {
13             nice+=prev;
14         } else {
15             nice += tmp;
16         }
17         prev = tmp;
18     }
19     std::cout << nice;
20     return 0;
21 }
```

[Отправить](#)[Предыдущая](#)[Следующая](#)