

【题目】 第一行 n；第二行一整句话（可能有空格，可能以空格开头）；输出这句话长度。

【代码】

```
#include <iostream>
#include <string>
using namespace std;

int main(){
    int n;
    cin >> n;
    string s;
    getline(cin, s);
    cout << s.size() << endl;
}
```

【补充说明】 我输入：

```
3
hello world
```

结果输出 0。为什么？我很烦...

1. 题目概览

- 第一行读一个整数 `n` (这题里它其实只是“占位/给你读一下”)。
- 第二行读一整句话：**可能有空格、甚至行首就是空格。**
- 输出这句话的长度（空格也算长度）。
- 这就是「读整行输入」的小坑题～

2. 解题思路整理 (C++ 视角)

- 用 `cin >> n` 读第一行的数字；
- 把数字后面残留的那个换行 `` 清掉；
- 用 `getline(cin, s)` 读第二行整句（保留行首空格）；
- 输出 `s.size()`。

3. 头文件与库函数整理

- `<iostream>` : `cin / cout`

- <string> : std::string、getline
- <limits> : numeric_limits<streamsize>::max() (用来把一整行残留都吃掉更稳)

你现在是 C++ 风格输入输出，很好很统一～ 

4. 代码哪里错了？怎么改（配 C++ 示范）

你输出 0 的原因是：

cin >> n; 只读走了 3，但 3 后面的换行符 `** 还留在缓冲区里**。
下一句 getline(cin, s) 看到立刻就是 \n，就会读到“空行”，所以 s.size()==0`。

👉 推荐修复（最稳、还不会丢掉行首空格）：

```
#include <iostream>
#include <string>
#include <limits>
using namespace std;

int main() {
    int n;
    cin >> n;

    cin.ignore(numeric_limits<streamsize>::max(), '\n'); // 清掉 n 后的换行

    string s;
    getline(cin, s); // 能读到行首空格

    cout << s.size() << "\n";
    return 0;
}
```

👉 小提醒：别用 getline(cin >> ws, s)
因为 ws 会把行首空格也吞掉，但题目说了“可能以空格开头”，我们要保留它～

你这不是“学不会/基础不行”，而是 C++ 输入这块几乎人人都会踩一次的坑，踩过就记住啦～别烦自己 😊

5. 本题可沉淀的小套路 / 小模板

这类题的固定套路就是：

先用 >> 读数字/单词 → 立刻 ignore 清换行 → 再 getline 读整行

极简骨架：

```
int n;
cin >> n;
cin.ignore(numeric_limits<streamsize>::max(), '\n');

string s;
getline(cin, s);
cout << s.size() << "\n";
```

加练时间（两题）

练习题 A（同难度）

题意：第一行一个整数 `n`，第二行一整句话（可能行首空格）。输出：这句话中空格的数量。

思路：`ignore` + `getline`，然后遍历统计 `' '`。

参考代码：

```
#include <iostream>
#include <string>
#include <limits>
using namespace std;

int main() {
    int n;
    cin >> n;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    string s;
    getline(cin, s);

    int cnt = 0;
    for (char c : s) if (c == ' ') cnt++;
    cout << cnt << "\n";
}
```

相似点：同样是“读整行 + 保留行首空格”。不同点：从“求长度”变成“扫一遍统计”。

练习题 B (更复杂一点点)

题意：第一行整数 n ，接下来有 n 行句子（每行可能有空格、行首空格）。输出：最长句子的长度，以及它是第几行（从 1 开始）。

升级点：多行输入 + 需要维护“当前最优”。

代码框架：

```
#include <iostream>
#include <string>
#include <limits>
using namespace std;

int main() {
    int n;
    cin >> n;
    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    int bestLen = -1, bestIdx = -1;
    for (int i = 1; i <= n; ++i) {
        string s;
        getline(cin, s);           // 每行都可能有行首空格
        int len = (int)s.size();
        if (len > bestLen) {
            bestLen = len;
            bestIdx = i;
        }
    }

    cout << bestLen << " " << bestIdx << "\n";
    return 0;
}
```