

# while(cin) 输入字符串

---

## 人员

韩承睿、牟茗、刘嘉航、辛帅辰、高健桓、李翰如、方俊喆、崔吉诺、刘祺、夏硕承、齐振玮、谢亚锴、王静嘉、牛同泽、刘轩铜 到课

## 作业检查

韩承睿 已完成

牟茗 已完成

刘嘉航 已完成

辛帅辰 已完成

高健桓 已完成

李翰如 已完成

方俊喆 已完成

崔吉诺 已完成

刘祺 已完成

夏硕承 已完成

秦显森 未打卡

齐振玮 未打卡

谢亚锴 已完成

王静嘉 未打卡

牛同泽 已完成

徐浩然 已完成

刘轩铜 第一次进班

## 作业

必做: **noi 1.7 31:字符串p型编码** 和 **noi 1.7 10:简单密码**

选做: noi 1.7 的所有题目, 现在大家把字符串的相关知识都学了, 这些题目都可以做

下节课默写: 23:过滤多余的空格 25:最长最短单词 24:单词的长度 28:单词倒排 26:字符串最大跨距

## 课堂表现

默写过了的同学：辛帅辰、刘祺、崔吉诺、夏硕承、高健桓、牟茗、方俊喆、牛同泽、刘嘉航

两个都过了的同学：刘祺、崔吉诺、辛帅辰、牟茗、谢亚锴

有几位同学课上注意力不集中，需要老师多次提醒，希望以后可以改正

## 课堂内容

### noi 1.7 26:字符串最大跨距

找到两个逗号的位置 p1 和 p2，然后根据两个逗号截取出 s,s1,s2 三部分  
然后，用find在s中找到最左边的s1 和 最右边的s2  
然后求距离差即可

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    string str; cin >> str; str += ',';

    string s, s1, s2, p;
    int n = str.size();
    for (int i = 0; i < n; ++i) {
        if (str[i] == ',') {
            if (s == "") s = p;
            else if (s1 == "") s1 = p;
            else s2 = p;
            p = "";
        } else p += str[i];
    }

    int b1 = s.find(s1), b2 = s.rfind(s2);
    if (b1 == -1 || b2 == -1) {
        cout << -1 << endl;
        return 0;
    }

    int s1R = b1 + (int)s1.size() - 1;
    int s2L = b2;
    if (s2L - s1R - 1 >= 0) cout << s2L - s1R - 1 << endl;
    else cout << -1 << endl;
    return 0;
}
```

### noi 1.7 23:过滤多余的空格

找到每个空格，如果前面是空格，则不输出这个空格；如果前面不是空格，则输出这个空格

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string s;
    getline(cin, s);
    int n = s.size();
    for (int i = 0; i < n; i++) {
        if (s[i] == ' ') {
            if (s[i-1] != ' ') cout << " ";
        }
        else {
            cout << s[i];
        }
    }
    return 0;
}
```

### noi 1.7 27:单词翻转

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    string s, t;
    getline(cin, s);
    int n = s.size();
    for (int i = 0; i < n; i++) {
        if (s[i] != ' ') {
            t += s[i];
        }
        else {
            if (s[i-1] == ' ') {
                cout << " ";
            }
            else {
                reverse(t.begin(), t.end());
                cout << t;
                t = "";
                cout << " ";
            }
        }
    }

    if (t.size() > 0) {
```

```
        reverse(t.begin(), t.end());
        cout << t;
    }
    return 0;
}
```

## 新知识

```
#include <iostream>
#include <cstring>

using namespace std;

int main()
{
    char c;
    cin >> c;
    if (isdigit(c)) {
        cout << "是数字" << endl;
    }
    else {
        cout << "不是数字" << endl;
    }
    if (isalpha(c)) {
        cout << "是字母" << endl;
    }
    else {
        cout << "不是字母" << endl;
    }
    if (islower(c)) {
        cout << "是小写" << endl;
    }
    else {
        cout << "不是小写" << endl;
    }
    if (isupper(c)) {
        cout << "是大写" << endl;
    }
    else {
        cout << "不是大写" << endl;
    }
    return 0;
}
```

可以让我们一直输入，一直输入，到某一刻时，可以用一个信号(ctrl+Z)，来告诉我们输入停止了

```
while (true) {
```

```
    cin >> s;
}
```

### noi 1.7 23:过滤多余的空格

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string s;
    while (cin >> s) {
        cout << s << " ";
    }
    return 0;
}
```

### noi 1.7 25:最长最短单词

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string minn, maxx;
    for (int i = 0; i < 1000; i++) {
        minn += " ";
    }

    string s;
    while (cin >> s) {
        if (s.size() < minn.size()) {
            minn = s;
        }
        if (s.size() > maxx.size()) {
            maxx = s;
        }
    }

    cout << maxx << endl;
    cout << minn << endl;
    return 0;
}
```

## noi 1.7 24:单词的长度

先输出第一个长度，对于后面的每个数，前面输出逗号即可

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string s;
    bool flag = false;
    while (cin >> s) {
        if (flag == false) {
            cout << s.size();
            flag = true;
        }
        else {
            cout << "," << s.size();
        }
    }
    return 0;
}
```

## noi 1.8 28:单词倒排

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string s, t;
    while (cin >> s) {
        t = s + " " + t;
    }
    cout << t << endl;
    return 0;
}
```