

回文子串+字符串移位包含问题

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、秦显森、齐振玮、谢亚锴、王静嘉、牛同泽、徐浩然 到课

作业检查

韩承睿 已完成

牟茗 已完成

刘嘉航 未打卡

辛帅辰 已完成

高健桓 已完成

李翰如 已完成

方俊喆 未完成

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 上周请假

牛同泽 已完成

徐浩然 已完成

作业

必做: **noi 1.7 26:字符串最大跨距** 和 **noi 1.7 23:过滤多余的空格**

选做: **noi 1.7 27:单词翻转**

下周默写: **noi 1.7 34:回文子串** 和 **noi 1.7 19:字符串移位包含问题**

课堂表现

牟茗、刘祺、辛帅辰、徐浩然、崔吉诺、夏硕承、谢亚锴、方俊喆、刘嘉航、秦显森、李翰如 默写过了，提出表扬！！

有几位同学课上听讲不认真，被老师批评了，以后老师讲课时一定要好好听课，等听课结束后会给大家留时间写代码的。

课堂内容

noi 1.7 20:删除单词后缀

```
// 方法一
#include <iostream>
#include <cstring>
#include <string>
using namespace std;
int main() {
    string str;
    cin >> str;
    int len = str.size();
    if (str[len-2]=='e' && str[len-1]=='r') {
        for (int i = 0; i <= len-3; i++) {
            cout << str[i];
        }
    }
    else if (str[len-2]=='l' && str[len-1]=='y') {
        for (int i = 0; i <= len-3; i++) {
            cout << str[i];
        }
    }
    else if (str[len-3]=='i' && str[len-2]=='n' && str[len-1]=='g') {
        for (int i = 0; i <= len-4; i++) {
            cout << str[i];
        }
    }
    else {
        cout << str << endl;
    }
    return 0;
}
```

```
// 方法二
#include <iostream>
#include <cstring>
#include <string>
using namespace std;
int main() {
    string str;
    cin >> str;
    int len = str.size();
    if (str[len-2]=='e' && str[len-1]=='r') {
```

```

        cout << str.substr(0, len-2) << endl;
    }
    else if (str[len-2]=='l' && str[len-1]=='y') {
        cout << str.substr(0, len-2) << endl;
    }
    else if (str[len-3]=='i' && str[len-2]=='n' && str[len-1]=='g') {
        cout << str.substr(0, len-3) << endl;
    }
    else {
        cout << str << endl;
    }
    return 0;
}

```

```

// 方法三
#include <iostream>
#include <cstring>
#include <string>
using namespace std;
int main() {
    string str;
    cin >> str;
    int len = str.size();
    if (len>=2 && str.substr(len-2,3)=="er") {
        cout << str.substr(0, len-2) << endl;
    }
    else if (len>=2 && str.substr(len-2,3)=="ly") {
        cout << str.substr(0, len-2) << endl;
    }
    else if (len>=3 && str.substr(len-3,3)=="ing") {
        cout << str.substr(0, len-3) << endl;
    }
    else {
        cout << str << endl;
    }
    return 0;
}

```

noi 1.7 34:回文子串

1. 枚举长度, k: 从 2 到 len
2. 枚举起点, i: 从 0 到 len-k
3. 验证 k,i 对应的字符串是不是 回文 -> t = s.substr(i,k) -> 判断 t 是不是回文
4. reverse -> t2 = t, reverse(t) if (t == t2) { }

```

#include <iostream>
#include <string>

```

```

#include <algorithm>

using namespace std;

int main() {
    string str;
    cin >> str;
    int len = str.size();
    for (int k = 2; k <= len; k++) {
        for (int i = 0; i <= len-k; i++) {
            string t = str.substr(i,k);
            string t2 = t;
            reverse(t.begin(), t.end());
            if (t2 == t) {
                cout << t << endl;
            }
        }
    }
    return 0;
}

```

noi 1.7 19:字符串移位包含问题

1. 让第一个长，让第二个短
2. 只需要判断 第二个 是不是 第一个循环移位(0~len-1) 的一部分
3. k: 0~len-1


```

string t2 = t.substr(k) + t.substr(0,k)
if (t2.find(s)!=string::npos) {

}

```

```

#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

int main() {
    string s, t;
    cin >> s >> t;
    int lens = s.size(), lent = t.size();
    if (lens < lent) {
        string k = s;
        s = t;
        t = k;
        lens = s.size(), lent = t.size();
    }

    bool flag = false;

```

```
for (int k = 0; k < lens; k++) {
    string s2 = s.substr(k) + s.substr(0, k);
    if (s2.find(t) != string::npos) {
        flag = true;
        break;
    }
}

if (flag == true) {
    cout << "true" << endl;
}
else {
    cout << "false" << endl;
}
return 0;
}
```