

string

人员

韩承睿、牟茗、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、牛同泽、徐浩然 到课，刘嘉航 线上

作业检查

韩承睿 已完成

牟茗 已完成

辛帅辰 已完成

高健桓 已完成

方俊喆 已完成

崔吉诺 已完成

刘祺 已完成

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

刘嘉航 未打卡

李翰如 请假

秦显森 已完成

夏硕承 已完成

作业

必做：noi 1.7 18:验证子串 和 noi 1.7 20:删除单词后缀

选做：noi 1.7 11:潜伏者

课堂表现

同学们课上听讲都比较认真，不过今天讲的题目有些复杂，课上没有让同学们实现，所以同学们课下要好好复习

课堂内容

noi 1.7 16:忽略大小写的字符串比较

```

#include <iostream>
#include <string>

using namespace std;

int main()
{
    string a, b;
    getline(cin, a);
    getline(cin, b);
    int lena = a.size(), lenb = b.size();
    for (int i = 0; i < lena; i++) {
        if (a[i] >= 'a' && a[i] <= 'z') {
            a[i] = a[i] - 'a' + 'A';
        }
    }
    for (int i = 0; i < lenb; i++) {
        if (b[i] >= 'a' && b[i] <= 'z') {
            b[i] = b[i] - 'a' + 'A';
        }
    }

    if (a < b) {
        cout << "<";
    }
    else if (a > b) {
        cout << ">";
    }
    else {
        cout << "=";
    }
    return 0;
}

```

noi 1.7 14:大小写字母互换

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

using namespace std;

int main()
{
    string s;
    getline(cin, s);
    int n = s.size();
    for (int i = 0; i < n; ++i) {
        if (s[i] >= 'A' && s[i] <= 'Z') s[i] = s[i] - 'A' + 'a';
        else if (s[i] >= 'a' && s[i] <= 'z') s[i] = s[i] - 'a' + 'A';
    }
    cout << s << endl;
    return 0;
}
```

noi 1.7 17:字符串判等

```

#include <iostream>
#include <string>

using namespace std;

int main()
{
    string a, b;
    getline(cin, a);
    getline(cin, b);
    int lena = a.size(), lenb = b.size();

    string s, t;
    for (int i = 0; i < lena; i++) {
        if (a[i] != ' ') {
            if (a[i] >= 'a' && a[i] <= 'z') {
                a[i] = a[i] - 'a' + 'A';
            }
            s += a[i];
        }
    }
    for (int i = 0; i < lenb; i++) {
        if (b[i] != ' ') {
            if (b[i] >= 'a' && b[i] <= 'z') {
                b[i] = b[i] - 'a' + 'A';
            }
            t += b[i];
        }
    }

    if (s == t) {
        cout << "YES" << endl;
    }
    else {
        cout << "NO" << endl;
    }
    return 0;
}

```

noi 1.7 18:验证子串

```

1. s(lens) t(lent)
2. 根据 lens/lent 的大小关系，确定 怎么判断
   if (lens < lent) {
       判 s 是不是 t 的子串
   }
   else if (lens > lent) {
       .. t .... s ....
   }
   else {
       s == t
   }
3. lens < lent:
   看 t 有没有一段跟 s 一模一样的
   t[0] --- t[lens-1]
   t[1] --- t[lens]
   ...
   t[lent-lens] --- t[lent-1]

   for (int l=0, r=lens-1; l<=lent-lens; l++, r++) {
       // 判断 l~r 这一部分跟 s 是不是一模一样
   }
4. 判断 l~r 这一部分跟 s 是不是一模一样
   t: l ~ r
   s: 0 ~ lens-1
   t[l] ~ s[0]
   t[l+1] ~ s[1]
   ...
   t[r] ~ s[lens-1]

   for (int i=l, j=0; i<=r; i++, j++) {
       // 判断 s[i] 和 t[j] 相不相等
   }

```

```

#include <iostream>
#include <string>

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 flag2 = false;
    for (int l=0, r=lens-1; l<=lent-lens; l++, r++) {
        bool flag = true;
        for (int i=l, j=0; i<=r; i++, j++) {
            if (t[i] != s[j]) {
                flag = false;
                break;
            }
        }
        if (flag == true) {
            // 成功
            flag2 = true;
            break;
        }
    }

    if (flag2 == true) {
        cout << s << " is substring of " << t << endl;
    }
    else {
        cout << "No substring" << endl;
    }
    return 0;
}

```

string 的新方法

1. reverse

2. find

3. rfind

4. substr

```
string s;  
reverse(s.begin(), s.end()); // 翻转 s  
s.find("abc"); // 在 s 中从左到右寻找 "abc" 字符串，如果找到，返回首字母下标；如果没找到，返回 string::npos，即 int 下的 -1  
s.rfind("abc"); // 在 s 中从右到左寻找 "abc" 字符串  
string t = s.substr(7,3); // 从 s 的位置 7 开始，截取长度为 3 的字符串  
string t = s.substr(7); // 从 s 的位置 7 开始，截取剩余的全部字符串部分
```