

输入输出重定向

输入输出重定向

- 如果输入数据很庞大，需要一次又一次的重新输入和调试时可使用
- C语言的重定向
 - freopen函数
- C++的重定向
 - streambuf* rdbuf () const;
 - streambuf* rdbuf (streambuf *)
- 借助操作系统的重定向

C重定向例子

16:忽略大小写的字符串比较

总时间限制：1000ms 内存限制：65536kB

描述

一般我们用strcmp可比较两个字符串的大小，比较方法为对两个字符串从前往后逐个字符相比较（按ASCII码值大小比较），直到出现不同的字符或遇到'\0'为止。如果全部字符都相同，则认为相同；如果出现不相同的字符，则以第一个不相同的字符的比较结果为准（注意：如果某个字符串遇到'\0'而另一个字符串还未遇到'\0'，则前者小于后者）。但在有些时候，我们比较字符串的大小时，希望忽略字母的大小，例如"Hello"和"hello"在忽略字母大小写时是相等的。请写一个程序，实现对两个字符串进行忽略字母大小写的大小比较。

输入

输入为两行，每行一个字符串，共两个字符串。（每个字符串长度都小于80）

输出

如果第一个字符串比第二个字符串小，输出一个字符"<"；
如果第一个字符串比第二个字符串大，输出一个字符">"；
如果两个字符串相等，输出一个字符"="。

样例输入

```
Hello, how are you?  
hello, How are you?
```

样例输出

```
=
```

<pre> #include <iostream> using namespace std; #include <cstdio> int main() { freopen("f:\\in.txt", "r", stdin); //freopen("f:\\out.txt", "w", stdout); char buff1[128] = { 0 }, buff2[128] = { 0 }; cin.getline(buff1, 128); cin.getline(buff2, 128); char *p1, *p2; p1 = buff1; p2 = buff2; char temp1, temp2; while (*p1 != 0 && *p2 != 0) { temp1 = (*p1 >= 'a' && *p1 <= 'z') ? *p1 - 32 : *p1; temp2 = (*p2 >= 'a' && *p2 <= 'z') ? *p2 - 32 : *p2; if (temp1 < temp2) { cout << "<"; return 0; } else if (temp1 > temp2) { cout << ">"; return 0; } p1++; p2++; } if (*p1) { cout << ">"; return 0; } if (*p2) { cout << "<"; return 0; } cout << "="; return 0; } </pre>	<pre> #include <iostream> using namespace std; #include <fstream> int main() { ifstream ifile("f:\\in.txt"); stringstream *backup = cin.rdbuf(); cin.rdbuf(ifile.rdbuf()); char buff1[128] = { 0 }, buff2[128] = { 0 }; cin.getline(buff1, 128); cin.getline(buff2, 128); char *p1, *p2; p1 = buff1; p2 = buff2; char temp1, temp2; while (*p1 != 0 && *p2 != 0) { temp1 = (*p1 >= 'a' && *p1 <= 'z') ? *p1 - 32 : *p1; temp2 = (*p2 >= 'a' && *p2 <= 'z') ? *p2 - 32 : *p2; if (temp1 < temp2) { cout << "<"; return 0; } else if (temp1 > temp2) { cout << ">"; return 0; } p1++; p2++; } if (*p1) { cout << ">"; return 0; } if (*p2) { cout << "<"; return 0; } cout << "="; cin.rdbuf(backup); return 0; } </pre>
--	--