

```
#include <iostream>
    using namespace std;
 3 □ int main() {
        int i, length1, length2;
 4
        string s1, s2;
 6
        getline(cin,s1);
        getline(cin,s2);
 8
        length1 = s1.size();
 9
        length2 = s2.size();
10
        for (i = 0; i < length1; i++)</pre>
11
             if (s1[i] >= 'a' && s1[i] <= 'z')
12
                 s1[i] -= 'a' - 'A';
13
        for (i = 0; i < length2; i++)</pre>
             if (s2[i] >= 'a' && s2[i] <= 'z')
14
15
                 s2[i] -= 'a' - 'A';
16
        if (s1 == s2)
17
             cout << "=" << endl;
        else if (s1 > s2)
18
             cout << ">" << endl;</pre>
19
20
        else
             cout << "<" << endl;</pre>
21
22
        return 0;
23 L
```

1 识别变量

常见变量名 翻译循环变量 根据变量名的英文推断

2 找出关键语句

控制结构(for, if) 常见算法的基本操作 函数参数、返回值

3 理解代码段作用

翻译解释代码段

```
#include <iostream>
    using namespace std;
 3 □ int main() {
        int i, length1, length2;
 4
        string s1, s2;
 6
        getline(cin,s1);
 7
        getline(cin,s2);
 8
        length1 = s1.size();
 9
        length2 = s2.size();
10
        for (i = 0; i < length1; i++)</pre>
11
             if (s1[i] >= 'a' && s1[i] <= 'z')
12
                 s1[i] -= 'a' - 'A';
        for (i = 0; i < length2; i++)</pre>
13
            if (s2[i] >= 'a' && s2[i] <= 'z')
14
15
                 s2[i] -= 'a' - 'A';
16
        if (s1 == s2)
17
            cout << "=" << endl;
18
        else if (s1 > s2)
19
            cout << ">" << endl;
20
        else
            cout << "<" << endl;</pre>
21
22
        return 0;
23 L
```

解释变量的作用

s1	输入第一行的字符串
s2	输入第二行的字符串
length1	s1的长度
length2	s2的长度

```
#include <iostream>
    using namespace std;
 3 □ int main() {
        int i, length1, length2;
 4
        string s1, s2;
 6
        getline(cin,s1);
 7
        getline(cin,s2);
 8
        length1 = s1.size();
 9
        length2 = s2.size();
10
        for (i = 0; i < length1; i++)</pre>
11
             if (s1[i] >= 'a' && s1[i] <= 'z')
12
                 s1[i] -= 'a' - 'A';
13
        for (i = 0; i < length2; i++)</pre>
             if (s2[i] >= 'a' && s2[i] <= 'z')
14
15
                 s2[i] -= 'a' - 'A';
16
        if (s1 == s2)
17
             cout << "=" << endl;
        else if (s1 > s2)
18
19
             cout << ">" << endl;
20
        else
             cout << "<" << endl;</pre>
21
22
        return 0;
23 L
```

关键语句

遍历字符串s1

将小写字符s1[i]转为大写

遍历字符串s2

将小写字符s2[i]转为大写

比较s1和s2的字典序

```
1 #include <iostream>
    using namespace std;
 3 □ int main() {
        int i, length1, length2;
        string s1, s2;
        getline(cin,s1);
 6
 7
        getline(cin,s2);
 8
        length1 = s1.size();
        length2 = s2.size();
        for (i = 0; i < length1; i++)</pre>
10
             if (s1[i] >= 'a' && s1[i] <= 'z')
11
                 s1[i] -= 'a' - 'A';
12
        for (i = 0; i < length2; i++)</pre>
13
             if (s2[i] >= 'a' && s2[i] <= 'z')
14
                 s2[i] -= 'a' - 'A';
15
16
        if (s1 == s2)
17
             cout << "=" << endl;
        else if (s1 > s2)
18
             cout << ">" << endl;</pre>
19
        else
20
             cout << "<" << endl;</pre>
21
22
        return 0;
23 L }
```

将第10行修改为for (i = 1; i <= length1; i++), 程序运行会出错。()

```
1 #include <iostream>
    using namespace std;
 3 □ int main() {
         int i, length1, length2;
        string s1, s2;
        getline(cin,s1);
 6
 7
        getline(cin,s2);
 8
        length1 = s1.size();
        length2 = s2.size();
        for (i = 0; i < length1; i++)</pre>
10
             if (s1[i] >= 'a' && s1[i] <= 'z')
11
12
                 s1[i] -= 'a' - 'A';
         for (i = 0; i < length2; i++)</pre>
13
             if (s2[i] >= 'a' && s2[i] <= 'z')
14
                 s2[i] -= 'a' - 'A';
15
16
         if (s1 == s2)
             cout << "=" << endl;</pre>
17
        else if (s1 > s2)
18
             cout << ">" << endl;</pre>
19
20
        else
             cout << "<" << endl;</pre>
21
22
        return 0;
23 L }
```

```
1 #include <iostream>
    using namespace std;
 3 □ int main() {
         int i, length1, length2;
        string s1, s2;
 5
        getline(cin,s1);
 6
        getline(cin,s2);
 7
 8
        length1 = s1.size();
        length2 = s2.size();
 9
        for (i = 0; i < length1; i++)</pre>
10
             if (s1[i] >= 'a' && s1[i] <= 'z')
11
                 s1[i] -= 'a' - 'A';
12
         for (i = 0; i < length2; i++)</pre>
13
             if (s2[i] >= 'a' && s2[i] <= 'z')
14
                 s2[i] -= 'a' - 'A';
15
16
         if (s1 == s2)
             cout << "=" << endl;</pre>
17
        else if (s1 > s2)
18
             cout << ">" << endl;</pre>
19
20
        else
             cout << "<" << endl;</pre>
21
22
        return 0;
23 L }
```

将第6行改为cin>>s1, 第7行改为cin>>s2, 程序运行结果不变。 ()

```
1 #include <iostream>
    using namespace std;
 3 □ int main() {
         int i, length1, length2;
        string s1, s2;
        getline(cin,s1);
 6
 7
        getline(cin,s2);
 8
        length1 = s1.size();
        length2 = s2.size();
        for (i = 0; i < length1; i++)</pre>
10
             if (s1[i] >= 'a' && s1[i] <= 'z')
11
12
                 s1[i] -= 'a' - 'A';
         for (i = 0; i < length2; i++)</pre>
13
             if (s2[i] >= 'a' && s2[i] <= 'z')
14
                 s2[i] -= 'a' - 'A';
15
16
         if (s1 == s2)
             cout << "=" << endl;</pre>
17
        else if (s1 > s2)
18
             cout << ">" << endl;</pre>
19
20
        else
             cout << "<" << endl;</pre>
21
22
        return 0;
23 L }
```

```
1 #include <iostream>
 2 using namespace std;
 3 □ int main() {
        int i, length1, length2;
 4
 5
        string s1, s2;
 6
        getline(cin,s1);
 7
        getline(cin,s2);
 8
        length1 = s1.size();
        length2 = s2.size();
        for (i = 0; i < length1; i++)</pre>
10
             if (s1[i] >= 'a' && s1[i] <= 'z')</pre>
11
                 s1[i] -= 'a' - 'A';
12
        for (i = 0; i < length2; i++)</pre>
13
             if (s2[i] >= 'a' && s2[i] <= 'z')
14
15
                 s2[i] -= 'a' - 'A';
        if (s1 == s2)
16
             cout << "=" << endl;</pre>
17
        else if (s1 > s2)
18
             cout << ">" << endl;</pre>
19
20
         else
21
             cout << "<" << endl;</pre>
22
        return 0;
23 L }
```

选择 输入第一行"I have a dream.",第12行共执行() 次。

A. 8 B. 9 C. 10 D. 11

```
1 #include <iostream>
 2 using namespace std;
 3 □ int main() {
        int i, length1, length2;
 5
        string s1, s2;
 6
        getline(cin,s1);
        getline(cin,s2);
 8
        length1 = s1.size();
        length2 = s2.size();
        for (i = 0; i < length1; i++)</pre>
10
             if (s1[i] >= 'a' && s1[i] <= 'z')</pre>
11
                 s1[i] -= 'a' - 'A';
12
        for (i = 0; i < length2; i++)</pre>
13
             if (s2[i] >= 'a' && s2[i] <= 'z')
14
                 s2[i] -= 'a' - 'A';
15
        if (s1 == s2)
16
             cout << "=" << endl;</pre>
17
        else if (s1 > s2)
18
             cout << ">" << endl;</pre>
19
20
         else
21
             cout << "<" << endl;</pre>
22
        return 0;
23 L }
```

先择 输入第一行"I have a dream.",输入第二行" I Have A Dream.",程序的输出是()。

A. = B. > C. < D. 无输出

1 识别变量

常见变量名 翻译循环变量 根据变量名的英文推断

2 找出关键语句

控制结构(for, if) 常见算法的基本操作 函数参数、返回值

3 理解代码段作用

翻译解释代码段

```
解释变量的作用
   #include<cstdio>
                                                   输入字符串
                                       st
   char st[100];
3 pint main() {
                                             关键语句
4
       scanf("%s", st);
       for (int i = 0; st[i]; ++i) {
5 
                                                 遍历字符串st
           if ('A' <= st[i] && st[i] <= 'Z')</pre>
6
               st[i] += 1;
                                              将大写字符st[i]改写
8
                                                   为st[i]+1
9
       printf("%s\n", st);
10
       return 0;
```

判断 输入的字符只能由大写字母或小写字母组成()。

判断 程序第5行++i改写成i++,程序运行结果不变()。

判断 程序的输入和输出一定不同()。

判断 输入全部是小写,输出结果与输入结果相同()。

```
选择 输入QuanGuoLianSai,输出是( )。

A. QuanGuoLianSai B. quanguoliansai C. RuanHuoMianTai D. QvboGvpLjboSbj
```

```
选择 输入Z,输出是 ( )
A. Z B. A C. z D. [
```

```
#include<iostream>
    using namespace std;
 3 □ int main() {
         int t[256];
 4
         string s;
         int i;
 6
         cin>>s;
 8
         for (i = 0; i < 256; i++)
 9
             t[i] = 0;
         for(i = 0; i < s.length(); i++)</pre>
10
             t[s[i]]++;
11
         for (i = 0; i < s.length(); i++)</pre>
12
13 申
             if (t[s[i]]== 1) {
             cout << s[i] << endl;</pre>
14
15
             return 0;
16
17
         cout<< "no" << endl;
18
         return 0;
19 <sup>L</sup> }
```

1 识别变量

常见变量名 翻译循环变量 根据变量名的英文推断

2 找出关键语句

控制结构(for, if) 常见算法的基本操作 函数参数、返回值

3 理解代码段作用

翻译解释代码段

#include<iostream> using namespace std; 3 □ int main() { int t[256]; 4 5 string s; 6 int i; 7 cin>>s; 8 for (i = 0; i < 256; i++)9 t[i] = 0;for(i = 0; i < s.length(); i++)</pre> 10 t[s[i]]++; 11 12 for (i = 0; i < s.length(); i++)</pre> **13** □ **if** (t[s[i]]== 1) { 14 cout << s[i] << endl;</pre> 15 return 0; 16 17 cout<< "no" << endl;</pre> return 0; 18 19 ^L }

解释变量的作用

t[]	计数器数组
S	输入的字符串
t[i]	ASCII值为i的字符出现次 数

```
#include<iostream>
    using namespace std;
 3 □ int main() {
         int t[256];
 4
         string s;
 6
         int i;
         cin>>s;
 8
         for (i = 0; i < 256; i++)
 9
             t[i] = 0;
         for(i = 0: i < s.length(); i++)</pre>
10
11
             t[s[i]]++;
12
         for (i = 0; i < s.length(); i++)
13 □
             if (t[s[i]]== 1) {
14
             cout << s[i] << endl;</pre>
15
             return 0;
16
17
         cout<< "no" << endl;
         return 0;
18
19 <sup>L</sup> }
```

怎样理解字符作为 数组下标

字符类型的ASCII 代码数值0-127

将字符的数值作为 下标

```
#include<iostream>
                                                   关键语句
   using namespace std;
3 □ int main() {
        int t[256];
4
        string s;
        int i;
6
        cin>>s;
        for (i = 0; i < 256; i++)
8
                                                  计数器数组清零
9
            t[i] = 0;
        for(i = 0; i < s.length(); i++)</pre>
10
                                              统计字符s[i]出现的次数
11
            t[s[i]]++;
12
        tor (1 = 0; 1 < s.lengtn(); 1++)</pre>
            if (t[s[i]]== 1) {
13 □
                                             按照字符串的顺序输出第
14
            cout << s[i] << endl;</pre>
                                               一个出现一次的字符
15
            return 0;
16
        cout<< "no" << endl;
17
18
        return 0;
19 <sup>∟</sup> }
```

```
1 #include<iostream>
    using namespace std;
 3 □ int main() {
         int t[256];
 4
         string s;
         int i;
 6
         cin>>s;
         for (i = 0; i < 256; i++)
 9
             t[i] = 0;
         for(i = 0; i < s.length(); i++)</pre>
10
11
             t[s[i]]++;
12
         for (i = 0; i < s.length(); i++)</pre>
             if (t[s[i]]== 1) {
13 □
14
             cout << s[i] << endl;</pre>
             return 0;
15
16
         cout<< "no" << endl;
17
18
         return 0;
19 <sup>L</sup> }
```

判断

8、9两行代码在程序中没有实际作用,可以删除,不影响运行结果()

```
1 #include<iostream>
 2 using namespace std;
 3 □ int main() {
        int t[256];
 4
        string s;
        int i;
 6
        cin>>s;
 8
        for (i = 0; i < 256; i++)
 9
            t[i] = 0;
        for(i = 0; i < s.length(); i++)</pre>
10
11
             t[s[i]]++;
12
        for (i = 0; i < s.length(); i++)</pre>
             if (t[s[i]]== 1) {
13 □
14
             cout << s[i] << endl;</pre>
             return 0;
15
16
        cout<< "no" << endl;
17
18
        return 0;
19 L }
```

判断

如果输入的字符串长度超过256, 第11行会数组越界()

```
1 #include<iostream>
 2 using namespace std;
 3 □ int main() {
        int t[256];
 4
        string s;
        int i;
 6
        cin>>s;
 8
        for (i = 0; i < 256; i++)
 9
            t[i] = 0;
        for(i = 0; i < s.length(); i++)</pre>
10
11
             t[s[i]]++;
12
        for (i = 0; i < s.length(); i++)</pre>
             if (t[s[i]]== 1) {
13 □
14
             cout << s[i] << endl;</pre>
15
             return 0;
16
        cout<< "no" << endl;
17
18
        return 0;
19 L }
```

判断 当输入的字符串只包含数字时,程序总是输出no()

```
1 #include<iostream>
    using namespace std;
 3 □ int main() {
        int t[256];
 4
        string s;
        int i;
 6
        cin>>s;
 8
        for (i = 0; i < 256; i++)
 9
            t[i] = 0;
        for(i = 0; i < s.length(); i++)</pre>
10
11
             t[s[i]]++;
12
        for (i = 0; i < s.length(); i++)</pre>
             if (t[s[i]]== 1) {
13 □
14
             cout << s[i] << endl;</pre>
15
             return 0;
16
        cout<< "no" << endl;
17
18
        return 0;
19 L }
```

判断 无论输入,输出要么只有1个字符,要么输出no()

```
1 #include<iostream>
 2 using namespace std;
 3 □ int main() {
         int t[256];
 5
         string s;
         int i;
 6
        cin>>s;
 8
         for (i = 0; i < 256; i++)
             t[i] = 0;
10
         for(i = 0; i < s.length(); i++)</pre>
11
             t[s[i]]++;
12
         for (i = 0; i < s.length(); i++)</pre>
13 □
             if (t[s[i]]== 1) {
             cout << s[i] << endl;</pre>
14
15
             return 0;
16 -
17
         cout<< "no" << endl;</pre>
         return 0;
18
19 <sup>L</sup> }
```

选择 输入yazxaz,程序输出结果()。

A.a B.x C.y D.z

```
1 #include<iostream>
    using namespace std;
 3 □ int main() {
         int t[256];
         string s;
         int i;
 6
         cin>>s;
 8
         for (i = 0; i < 256; i++)
             t[i] = 0;
         for(i = 0; i < s.length(); i++)</pre>
10
11
             t[s[i]]++;
         for (i = 0; i < s.length(); i++)</pre>
12
13 □
             if (t[s[i]]== 1) {
14
             cout << s[i] << endl;</pre>
15
             return 0;
16 -
17
         cout<< "no" << endl;</pre>
         return 0;
18
19 <sup>L</sup> }
```

选择 下面哪个输入,程序输出no()

A. abcdbca

B. http

C. www

D. happy

求字符串的逆序

下面的程序的功能是输入若干行字符串,每输入一行, 就按逆序输出该行,最后键入-1终止程序。请将程序 补充完整。

手算样例

输入

CDEF DFVC -1

输出

FEDC CVFD

输入

QWERTYUIOP ASDFGHJKL -1 ZXCVBNM

输出

POIUYTREWQ LKJHGFDSA

```
#include <iostream>
    #include <cstring>
    using namespace std;
    int maxline=200,kz;
    int reverse(char s[])
 6 □ {
 7
         int i,j,t;
         for(i=0,j=strlen(s)-1; i<j;--1--,--2--)
 9 🖨
10
             t=s[i]; s[i]=s[j]; s[j]=t;
11
12
         return 0;
13 L }
14 □ int main(){
15
         char line[100];
         cout<<"continue? -1 for end."<<endl;</pre>
16
17
         cin>>kz;
18
        while(--3--)
19 🗎
             cin>>line;
20
             --4--;
21
22
             cout<<li>cout<<endl;
             cout<<"continue? -1 for end."<<endl;</pre>
23
             cin>>kz:
24
25
26
         return 0;
27
```

识别变量

常见变量名 翻译循环变量 根据变量名的英文推断

2 找出关键语句

控制结构(for, if) 常见算法的基本操作 函数参数、返回值

3 理解代码段作用

翻译解释代码段

```
#include <iostream>
                                                strlen()
   #include <cstring>
                                                                求字符串长度
    using namespace std;
                                                 string
                                                                  length
    int maxline=200,kz;
    int reverse(char s[])
                                                字符串
                                                                   长度
6 □ {
 7
        int i,j,t;
        for(i=0,j=strlen(s)-1; i<j;--1--,--2--)</pre>
 9 🖨
                                                            控制变量,程序是否继续
10
            t=s[i]; s[i]=s[j]; s[j]=t;
                                                   kz
                                                                    循环
11
12
        return 0;
13 L }
                                                              反转,将字符串反转
                                               reverse()
14 □ int main(){
15
        char line[100];
        cout<<"continue? -1 for end."<<endl;</pre>
16
                                                                 左侧字符下标
17
        cin>>kz;
18
        while(--3--)
                                                                 右侧字符下标
19 \Box
            cin>>line;
20
21
            --4--;
                                                                输入一行字符串
                                                  line
22
            cout<<li>cout<<endl;
            cout<<"continue? -1 for end."<<endl;</pre>
23
24
            cin>>kz;
25
26
        return 0;
27
```

```
#include <iostream>
    #include <cstring>
    using namespace std;
    int maxline=200,kz;
    int reverse(char s[])
6 □ {
 7
         int i.i.t:
         for(i=0,j=strlen(s)-1; i<j;--1--,--2--)</pre>
 9 🖨
             t=s[i]; s[i]=s[j]; s[j]=t;
10
11
12
         return 0;
13 L }
14 □ int main(){
15
         char line[100];
         cout<<"continue? -1 for end."<<endl;</pre>
16
         cin>>kz:
17
18
         while(--3--)
19 🗎
             cin>>line;
20
             --4--;
21
22
             cout<<li>cout<<endl;
             cout<<"continue? -1 for end."<<endl;</pre>
23
             cin>>kz:
24
25
26
         return 0;
27
```

关键语句

从字符串两端依次交换编号i,j的字符,反转字符串s

交换变量s[i]和s[j]

输入-1结束程序 读入一个字符串并反转

(数字删除)

下面程序的功能是将字符串中的数字字符删除后输出。请填空。

手算样例

输入样例:

1He2llo3 World 456!

输出多少?

输出样例:

Hello World!

```
5分钟
```

```
完善程序
   #include <iostream>
   using namespace std;
                                                    识别变量
   const int SIZE = 30;
                                                常见变量名
4 □ int delnum(char *s) {
5
                                               翻译循环变量
       int i, j;
6
       j = 0;
                                            根据变量名的英文推断
       for (i = 0; s[i] !='\0'; i++)
           if (s[i]<'0' ____(1)___ s[i]>'9'){
8 E
              s[j] = s[i];
10
                                                  找出关键语句
11
12
                  (3)____;
       return
                                              控制结构(for, if)
13
                                            常见算法的基本操作
14 □ int main() {
                                             函数参数、返回值
15
       char s[SIZE];
       int len, i;
16
       cin.getline(s, sizeof(s));
17
       len = delnum(s);
18
       for (i = 0; i < len; i++)
19
                                             3
                                                 理解代码段作用
20
           cout << ____(4)____;
21
       cout << endl;
                                              翻译解释代码段
22
       return 0;
23
```

```
完善程序
   #include <iostream>
   using namespace std;
                                            delnum函数
   const int SIZE = 30;
   int delnum(char *s) {
                                          输入一个字符串
       int i, j;
                                           输出一个整数
       j = 0;
       for (i = 0; |s[i] !=' \0'; i++)
          if (s[i]<'0' ____
                             ___ s[i]>'9'){
8 E
              s[j] = s[i];
                                      c语言字符串以'\0'特殊字
10
                                         符表示字符串末尾
11
12
       return
                 (3) ;
13
14 □ int main() {
                                       s表示输入的字符数组
15
       char s[SIZE]; ←
16
       int len, i;
       cin.getline(s, sizeof(s));
17
                                      len表示去掉数字字符后
       len = delnum(s);
18
                                        所剩字符数组的长度
       for (i = 0; i < len; i++) \leftarrow
19
          cout << ____(4)____;
20
21
       cout << endl;
22
       return 0;
```

23

```
完善程序
    #include <iostream>
    using namespace std;
    const int SIZE = 30;
4 ☐ int delnum(char *s) {
        int i, j;
        j = 0;
        for (i = 0; s[i] !='\0'; i++)
            if (s[i]<'0' ____(1)___ s[i]>'9'){
                s[j] = s[i];
10
11
12
        return
                    (3)
13
14 □ int main() {
        char s[SIZE];
15
16
        int len, i;
        cin.getline(s, sizeof(s));
17
        len = delnum(s);
18
        for (i = 0; i < len; i++)
19
            cout << ____(4)____;
20
21
        cout << endl;
22
        return 0;
23
```

cin.getline函数 通过cin将一个字符串 输入到字符数组s中

第二个参数规定了最多可

允许读入 SIZE-1个字符

```
完善程序
    #include <iostream>
    using namespace std;
    const int SIZE = 30;
4 ☐ int delnum(char *s) {
5
        int i, j;
6
        j = 0;
        for (i = 0; s[i] !='\0'; i++)
            if (s[i]<'0' ____(1)___ s[i]>'9'){
8 E
                s[j] = s[i];
10
11
12
                    (3)_
        return
13
14 □ int main() {
15
        char s[SIZE];
        int len, i;
16
        cin.getline(s, sizeof(s));
17
        len = delnum(s);
18
        for (i = 0; i < len; i++)
19
20
            cout << ____(4)__
21
        cout << endl;
22
        return 0;
23
```

通过调用delnum函数 返回删除数字字符后所剩字符数 然后根据字符数进行循环 重新输出改动后的字符数组

```
完善程序
    #include <iostream>
    using namespace std;
    const int SIZE = 30;
4 □ int delnum(char *s) {
5
        int i, j;
6
        for (i = 0; s[i] !='\0'; i++)
            if (s[i]<'0' ____(1)___ s[i]>'9'){
8
                s[j] = s[i];
                    _(2)___;
10
11
12
        return
14 □ int main() {
        char s[SIZE];
15
16
        int len, i;
        cin.getline(s, sizeof(s));
17
        len = delnum(s);
18
19
        for (i = 0; i < len; i++)
            cout << ____(4)____;
20
21
        cout << endl;
22
        return 0;
23
```

对s中每个字符i 如果s[i]不是**数字字符** 就把该字符赋值到字串前部 也就是s[j]=s[i](j从0开始)

```
完善程序
   #include <iostream>
   using namespace std;
   const int SIZE = 30;
   int delnum(char *s) {
       int i, j;
6
       for (i = 0; s[i] !='\0'; i++)
          if (s[i]<'0' ____(1)___ s[i]>'9'){
              s[j] = s[i];
                                          传入字符数组s
10
                                          对s中每个字符i
11
                                        如果s[i]不是数字字符
12
       return
             (3) ;
                                      就把该字符赋到原字串前部
13
                                       结束后返回新字串长度i
14 □ int main() {
       char s[SIZE];
15
       int len, i;
                                      定义变量并cin输入字符串s
16
       cin.getline(s, sizeof(s));
17
       len = delnum(s);
18
       for (i = 0; i < len; i++)
19
          cout << ____(4)_
                                       返回得到新字符串长度
20
21
       cout << endl;
                                     并根据长度控制输出新字符串
22
       return 0;
```

23

```
完善程序
    #include <iostream>
    using namespace std;
    const int SIZE = 30;
    int delnum(char *s) {
        int i, j;
        j = 0;
        for (i = 0; s[i] !='\0'; i++)
8 E
            if (s[i]<'0' ____(1)___ s[i]>'9'){
                s[j] = s[i];
10
11
        return
12
13
14 □ int main() {
15
        char s[SIZE];
        int len, i;
16
        cin.getline(s, sizeof(s));
17
        len = delnum(s);
18
        for (i = 0; i < len; i++)
19
20
            cout << ____(4)____;
21
        cout << endl;
22
        return 0;
23
```

根据ASCII码表规律 判断不是数字字符的条件应是 s[i]<'0'或者s[i]>'9'

s[i]赋值给s[j]后,编号j往后移动

18行用整型len接收delnum返回值结合19行可知,返回值是原字串中去掉数字字符后的字符总数,也就是j

题目要求

重新输出去掉数字字符后的新字符串 所以用for依次输出其中每个元素s[i]

(字符串替换)给定一个字符串S(S仅包含大小写字母),下面的程序将S中的每个字母用规定的字母替换,并输出S经过替换后的结果。程序的输入是两个字符串,第一个字符串是给定的字符串S,第二个字符串S'由26个字母组成,它是az的任一排列,大小写不定,S'规定了每个字母对应的替换字母:S'中的第一个字母是字母A和a的替换字母,即S中的A用该字母的大写替换,S中的a用该字母的小写替换;S'中的第二个字母是字母B和b的替换字母,即S中的B用该字母的大写替换,S中的b用该字母的小写替换;……以此类推。

手算样例

输入

ABCD BCDEFGHIJKLMNOPQRSTUVWXYZA



BCDE



TYHGHKN QWERTYUIOPasdfghjklzxcvbnm



ZNIUIAF

手算样例

原文	Α	В	С	D	E	F	G	Н	I	G	K	L	М
密文	Q	W	Ш	R	Т	Υ	J	_	0	Р	Α	S	D
原文	N	0	Р	Q	R	S	Т	U	V	W	X	Y	Z
密文	F	G	Н	J	K	L	Z	Х	С	V	В	N	М

输入

SOeasy QWERTYUIOPasdfghjklzxcvbnm

输出

L G	t	q		n
-----	---	---	--	---

```
#include <iostream>
    #include <cstring>
    char change[26], str[5000];
    using namespace std;
    void CheckChangeRule()
 6 □ {
 7
         int i:
         for (i = 0; i < 26; i ++)
 9 🖨
10
11
12
13
    void ChangeString()
15 □ {
16
         int i;
17
         for (i = 0;i <strlen(str);i ++)</pre>
18 🖨
19
                      str[i] = change[str[i] - 'A'] -'a' + 'A';
20
21
               else
22
23
24
25
    int main()
26 ₽ {
27
         int i:
         cin >> str ;
28
29
        cin >> change;
         CheckChangeRule();
30
31
32
         cout << str << endl;</pre>
33
         return 0;
34
```

1 识别变量

常见变量名 翻译循环变量 根据变量名的英文推断

2 找出关键语句

控制结构(for, if) 常见算法的基本操作 函数参数、返回值

3 理解代码段作用

翻译解释代码段

```
#include <iostream>
   #include <cstring>
   char change[26], str[5000];
                                                                 26个元素,存储每个字符
                                                   change[]
   using namespace std;
                                                                       的替换规则
   void CheckChangeRule()
6 □ {
                                                                 输入的需要加密的字符串
                                                      str[]
 7
       int i:
       for (i = 0; i < 26; i ++)
                                    CheckChangeRule()
                                                                 检查替换规则,统一替换
10
                                                                  规则,大小写保持一致
11
12
13
   void ChangeString()
                                                                 修改字符串,对输入的字
15 □ {
                                          ChangeString()
16
       int i;
                                                                      符串进行替换
17
       for (i = 0;i <strlen(str);i ++)</pre>
18 🖨
19
                  str[i] = change[str[i] - 'A'] -'a' + 'A';
20
21
            else
22
23
24
25
   int main()
26 ₽ {
27
       int i:
       cin >> str ;
28
29
       cin >> change;
30
       CheckChangeRule();
31
32
       cout << str << endl;</pre>
33
       return 0;
34
```

```
#include <iostream>
    #include <cstring>
    char change[26], str[5000];
    using namespace std;
    void CheckChangeRule()
 6 🗐
         int i;
 7
         for (i = 0; i < 26; i ++)
 8
 9 ₫
10
             if (
                     change[i] -= 'A' - 'a';
11
12
13
14
    void ChangeString()
15 □ {
16
         int i;
17
         for (i = 0;i <strlen(str);i ++)</pre>
18 🖨
19
               if (
20
                      str[i] = change[str[i] - 'A'] -'a' +
21
               else
22
23
24
25
    int main()
26 □ {
27
         int i:
         cin >> str ;
28
29
        cin >> change;
30
         CheckChangeRule();
31
32
         cout << str << endl;</pre>
33
         return 0:
34
```

关键语句

统一将字符的替换规则用 小写字母表示

将大写转为小写

```
#include <iostream>
   #include <cstring>
                                                             关键语句
   char change[26], str[5000];
   using namespace std;
   void CheckChangeRule()
6 □ {
7
      int i:
      for (i = 0; i < 26; i ++)
9 🖨
10
                                                               当前要替换的第i个字符
                                                    str[i]
11
12
                                            change[str[i]]
                                                               当前要替换的第i个字符,
13
                                                              按照替换规则替换为哪个
   void ChangeString()
15 □ {
                                                                       字符
16
      int i;
17
      for (i = 0;i <strlen(str);i ++)</pre>
18 🖨
19
            if (
                                                          将change[x]转为大写
                 str[i] = change[str[i] - 'A'] -'a' +
20
21
            else
22
23
24
                                        change[str[i]-'A']表示str[i]是大写字符替换
25
   int main()
                                        后的字符。将小写转大写是因为替换规则
26 □ {
27
      int i:
                                        统一小写,而str[i]替换后要与原文大小写
      cin >> str ;
28
29
      cin >> change;
                                                            一致
30
      CheckChangeRule();
31
32
      cout << str << endl;</pre>
33
      return 0;
```

34

作业: 完善程序

231 字符串的逆序

1987 数字删除

697 字符串替换