**一、string的常见函数**

**1 string的查找和截取**

**1、相关函数**

**查找和截取：**

s.find(st)：在s字符串中寻找子串st第一次出现的下标（没有，判断返回值是-1）。

s.find(st,下标p)：在字符串s中从下标为p开始寻找子串st第一次出现的下标。

s.substr(下标p,长度len)：从字符串s下标为p开始截取len个字符。

s.substr(下标p)：从字符串下标为p开始一直截到最后。

**例子：find**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    string s = "this is my friend.";

    //查找：

    cout<<s.find("is")<<endl;

    cout<<s.find("is",3)<<endl;//从下标为3的位置开始找

//    cout<<s.find("your")<<endl;

    if(s.find("your") == -1){

        cout<<"NO"<<endl;

    }

    //截取：从字符串中拷贝一部分

    string t = s.substr(2,5);

    cout<<t<<endl;

    cout<<s.substr(2)<<endl;//如果没有截取长度，截取到最后

    return 0;

}

**例子：substr**

#include<bits/stdc++.h>

using namespace std;

int main(){

    string s = "that is your book";

    //截取函数：s.substr()

    //从下标为5开始截取，截2个字符

    string s2 = s.substr(5,7);

    cout<<s2<<endl;

    //从下标为8开始截取，截取到最后

    string s3 = s.substr(8);

    cout<<s3<<endl;

    return 0;

}

1116 - 调换位置

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | **,** | d | e |  |  |
| 0 | 1 | 2 | **3** | 4 | 5 | 6 | 7 |
|  |  |  | **p** |  | s.size()  -1 |  |  |

思路：将逗号之后的单词截取（substr）出来，将逗号之前的单词截取，交换位置输出。

#include<bits/stdc++.h>

using namespace std;

int main(){

  string s,s1,s2;

  cin>>s;

  //求逗号的下标

  int p = s.find(",");

  //逗号前面的子串

  s1 = s.substr(0,p);

  //逗号后面的子串，不需要描述截多长

  //截到最后

  s2 = s.substr(p+1);

  cout<<s2<<","<<s1;

    return 0;

}

1591 - 求子串的位置

请分别使用find及substr函数求解该问题；

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G | o |  | A | b | c |  | **g** | o | o | d |  | **g** | o | o | l | e | ! |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | **7** | 8 | 9 | 10 | 11 | **12** | 13 | 14 | 15 | 16 | 17 |

t = “go”

归纳：

如果t的长度为1，循环到s.size()-1

如果t的长度为2，循环到s.size()-2

循环到s.size()-t.size()的位置

**解法一：使用find求解**

#include<bits/stdc++.h>

using namespace std;

int main(){

    string s,t;//s:父字符串，t:子字符串

    getline(cin,s);

    getline(cin,t);

    //找到t在s中第一次出现的下标

    int p = s.find(t);

    if(p == -1) cout<<-1;

    else{

         //找出所有位置

         //当s中还能找到t时

         while(p != -1){

             cout<<p+1<<endl;//下标+1就是位置

             //从当前下标+1向后重新找

             p = s.find(t,p+1);

         }

    }

    return 0;

}

**解法二：使用substr求解**

#include<bits/stdc++.h>

using namespace std;

string s,t;

int main() {

    getline(cin,s);

    cin>>t;

    if(s.find(t) == -1){

        cout<<-1;

        return 0;

    }

    int l1 = s.size(),l2 = t.size();

    //循环到倒数第l2个字母

    for(int i = 0;i <= l1 - l2;i++){

        if(s.substr(i,l2) == t){

            cout<<i+1<<endl;

        }

    }

    return 0;

}

**注意：**

    string a = "this",b = "where";

cout<<a.size() - b.size()<<endl;

输出的结果不是-1，而是一个很大的整数。

因为xx.size()返回值是unsigned long long，unsigned表示无符号的，没有负数。

**如果字符串长度不是一个变化的值，尽量将长度赋给int再使用：**

    string a = "this",b = "where";

    int l1 = a.size(),l2 = b.size();

    cout<<l1 - l2;

**2、 string的删除**

**1、相关函数**

s.erase(开始下标p,删除长度len)：从下标p开始删除len个字符。

s.erase(开始下标p)：从下标p开始删到最后。

**例子：erase的使用**

    string s = "good hi study";

    s.erase(5,3);//删除 hi

    //s.erase(5)：从下标为5开始删除到最后

    cout<<s;

**2、例题讲解**

1131 - 删除指定字符

分别掌握假删除和真删除的实现；

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| c | a | b | a | a |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

**解法一：假删除**

#include <bits/stdc++.h>

using namespace std;

int main(){

    string s;

    char c;

    cin>>s>>c;

    //假删除：遍历每个字符，除了要删除的字符c以外，剩余的输出

    int i;

    for(i = 0;i < s.size();i++){

         if(s[i] != c){

             cout<<s[i];

         }

    }

}

**解法二：真删除**

#include <bits/stdc++.h>

using namespace std;

int main(){

    //sdf$$$sdf

    //当s中含有要删除的字符c时，找到c的位置，并删除

    string s;

    char c;

    cin>>s>>c;

    int p = s.find(c);

    //当s中有c时

    while(p != -1){

         //删除s中的c

         s.erase(p,1);

         //计算s中是否还有c

         p = s.find(c);

    }

    cout<<s;

}

**错误的解法：**

#include<bits/stdc++.h>

using namespace std;

/\*

abcccdccc

c

\*/

string s;

char c;

int main()

{

    cin>>s>>c;

    for(int i = 0;i < s.size();i++){

        if(s[i] == c) s.erase(i,1);

    }

    cout<<s;

    return 0;

}

说明：因为如果有连续的要删除的字符，删一个，后面的就移过来了，i++使得我们跳过了1个要删除的字符。

注意：不要在正在循环的字符串中删除或者插入，防止循环次数和预期不一致。

**3、string插入、替换函数**

（1）insert(下标，字符串s) ：在字符串下标为i的位置插入另一个字符串s；

（2）replace(i,len,str)：从下标为i开始，替换len个字符为str

#include<bits/stdc++.h>

using namespace std;

int main() {

    string s = "hello boy";

    s.insert(6,"good ");

    cout<<s<<endl;//hello good boy

    //替换

    s.replace(0,5,"hi");

    cout<<s<<endl;

}

1570 - 字符替换

分别掌握使用s[i]直接更改以及replace函数替换的实现；

**解法一：逐个判断是否是要替换的字符，如果是，则替换**

#include <bits/stdc++.h>

using namespace std;

int main(){

    //判断每个字符是否是要替换的字符

    //如果是，则直接替换成新字符

    string s;

    char c1,c2;

    cin>>s>>c1>>c2;

    int i;

    for(i = 0;i < s.size();i++){

         if(s[i] == c1){

             s[i] = c2;

         }

    }

    cout<<s;

}

**解法二：使用replace函数替换**

#include <bits/stdc++.h>

using namespace std;

int main(){

    //当s中含有字符c1时，替换为字符c2

    string s;

    string c1,c2;

    cin>>s>>c1>>c2;

    int p = s.find(c1);

    while(p != -1){

         s.replace(p,1,c2);

         p = s.find(c1);

    }

    cout<<s;

}

**二、is函数、to函数**

**请注意：以下这些函数，都不是字符串函数，不需要写s.xx()！！！！！！**

**1、相关函数**

**字符类型**判断函数：（**非string函数**）

isalpha(c):判断c是否为字母

islower(c):判断是否为小写

isupper(c):判断是否为大写

isdigit(c):判断是否为数字

说明：返回非0表示真，返回0表示假；

判断是字母：if(isalpha(c))

判断不是字母：if(!isalpha(c))

!表示取反，也就是如果原来判断的是真，那么加!表示判断假。

字符类型转换函数：（非string函数）

tolower(c)：字符转小写

toupper(c)：字符转大写

说明： 返回int，返回字母对应的ascii码 ；☆☆☆

例子：

    char c = 'a';

    //默认得到的是ascii码

    cout<<toupper(c)<<endl;//65

    //将ascii码转换回字符

    c = toupper(c);

    cout<<c<<endl;

排序和倒序函数：（非string函数）

sort(起始地址,结束地址+1)：数组升序排序

reverse(起始地址,结束地址+1)：数组逆序

获取头尾指针

s.begin()：获取字符串s的头位置（指针）

s.end()：获取字符串s的尾位置（最后一个字符后面的位置）（指针）

**字符串排序：**

    string s = "cbadfe";

    sort(s.begin(),s.end());

**字符串数组排序：默认按字典码排序**

    //定义数组，存放5个字符串

    string a[5];

    a[0] = "23";

    a[1] = "12";

    a[2] = "5";

    a[3] = "2";

    a[4] = "120";

    sort(a,a+5);

    for(int i = 0;i < 5;i++){

         cout<<a[i]<<endl;

    }

**注意：**

bool类型是一个表示真假的类型，真值为true，假值为false！

bool f = true;//直接赋值

bool f = 3 > 2;//赋值为判断式

bool f = 3 > 2 && 5 == 5;

true打印，结果为1，false打印结果为0

if判断、while判断、for判断，除了写判断式，也可以写整数，0表示假，非0表示真！

  int a = 5;

  if(a){

       cout<<"yes";

  }else{

       cout<<"no";

  }

**因此，不要出现下面的写法，下面的写法，if的判断永远为真！**

    int a = 0;

    if(**a = 5**){

         cout<<"yes"<<endl;

    }else{

         cout<<"no"<<endl;

    }

    cout<<a<<endl;

**例子：is函数的使用**

  char c;

  cin>>c;

  //包含大写字母或者小写字母

  //c >= 'a' && c <= 'z' || c >= 'A' && c <= 'Z'

  if(isalpha(c)){

       cout<<"字母"<<endl;

       if(islower(c)){

           cout<<"小写"<<endl;

       }else if(isupper(c)){

           cout<<"大写"<<endl;

       }

  //c >= '0' && c <= '9'

  }else if(isdigit(c)){

       cout<<"数字"<<endl;

  }

**例子：to函数的使用**

    char c;

    c = 'b';

    //将ascii码赋值给字符，获取编码对应的字符

    c = toupper(c);

    cout<<c<<endl;

**2、例题讲解**

1408 - 合法的变量名？

#include <bits/stdc++.h>

using namespace std;

int main(){

    string s;

    bool f = true;//假设合法

    getline(cin,s);

    //判断非法的情况

    //1、变量名含有除了字母数字下划线以外的字符

    for(int i = 0;i < s.size();i++){

         if((isalpha(s[i]) || isdigit(s[i]) || s[i] == '\_') == false){

             f = false;

             break;

         }

    }

    //2、以数字开头

    if(isdigit(s[0])){

         f = false;

    }

    //3、含有4个关键词

    if(s == "int" || s == "double" || s == "cin" || s == "cout"){

         f = false;

    }

    if(f){

         cout<<"yes";

    }else{

         cout<<"no";

    }

}

N. 元音字母转大写辅音字母转小写

#include<bits/stdc++.h>

using namespace std;

string s;

string t = "aeiouAEIOU";

int main() {

    getline(cin,s);

    //枚举每个字母，如果是元音字母转大写，其余字母转小写

    for(int i = 0;i < s.size();i++){

        if(isalpha(s[i])){

            //如果是元音

            if(t.find(s[i]) != -1){

                s[i] = toupper(s[i]);

            }else{

                s[i] = tolower(s[i]);

            }

        }

    }

    cout<<s;

    return 0;

}

**三、sort和reverse**

**例子：sort和reverse函数的使用**

**有a数组如下：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 2 | 1 | 5 | 4 |  |  |  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| a首地址 | 地址：  a+1 | 地址：  a+2 | 地址：  a+3 | 地址：  a+n-1 | 地址：  a+n |  |  |

**用sort对数组排序：（数组的本质是a[0]的地址）**

sort(数组的起始地址,数组的结束地址+1)：对数组所有元素排序！

reverse(开始地址,结束地址+1)：将数组颠倒！

#include <bits/stdc++.h>

using namespace std;

int main(){

    int a[5] = {3,2,1,5,4};

    int n = 5;

    sort(a,a + n);//对数组所有元素升序排序

    对数组第2个数~倒数第2个数升序排序

    sort(a+1,a+n-1);

    数组逆序函数

    reverse(a,a+n);

    for(int i = 0;i < n;i++){

         cout<<a[i]<<" ";

    }

}

**注意：数组的本质是数组中下标为0的元素的地址；（首元素的地址）**

**例子：使用sort对字符串的字符进行排序！**

    string s = "cabed";

    sort(s.begin(),s.end());//s.begin()：开始地址  s.end()：结束地址+1

    cout<<s;

**2、例题讲解**

1167 - 字典排序

#include <bits/stdc++.h>

using namespace std;

string a[20];

int n;

int main(){

    cin>>n;

    for(int i = 1;i <= n;i++){

        cin>>a[i];

    }

    sort(a+1,a+n+1);

    for(int i = 1;i <= n;i++){

       cout<<a[i]<<" ";

    }

    return 0;

}

1117 - 重新排列

#include <bits/stdc++.h>

using namespace std;

int main(){

    /\*

      对20位整数的每一位从小到大排序

      如果第一个元素为'0'，则字符串中第一个非'0'交换

      231005 -> 001235 -> 100235

    \*/

    string s;

    cin>>s;

    sort(s.begin(),s.end());

    char t;//用于变量的交换

    //如果首元素为'0'，则将其与第一个非'0'元素交换

    if(s[0] == '0'){

         for(int i = 0;i < s.size();i++){

             if(s[i] != '0'){

//               t = s[0];

//               s[0] = s[i];

//               s[i] = t;

                 swap(s[0],s[i]);

                  break;

             }

         }

    }

    cout<<s;

}

**注意：**

**swap(a,b)：交换变量a、b的值；**