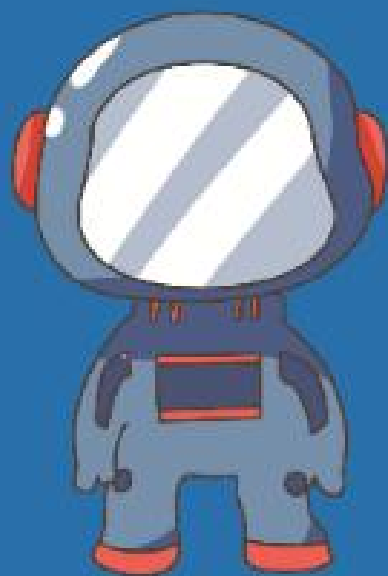




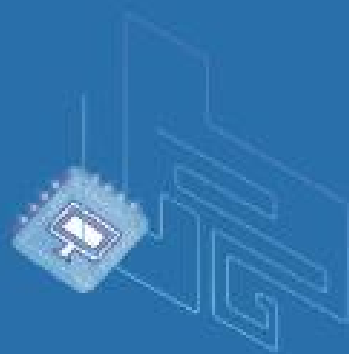
学习目标



1. 掌握字符串四个常用函数用法
2. 掌握字符串模拟经典题型



知识讲授





字符串常用函数

<code>s.insert(pos,s2)</code>	在s下标为pos的位置插入string类型s2
<code>s.erase(pos,len)</code>	删除s中下标为pos开始的len个字符
<code>s.substr(pos,len)</code>	截取s中下标为pos起的len个字符
<code>s.find(s2,pos)</code>	在s中以pos位置起查找s2第一次出现的位置

insert插入函数

- 格式：字符串1.insert(pox,字符串2);
- 功能：在字符串1中的index下标对应的元素之前插入字符串2。

```
string a="haha",b="123";
```

```
a.insert(2,b);
```

```
cout<<a;
```

出现结果是：

ha123ha



erase删除函数

- 格式：字符串1.erase(pos,len);
- 功能：删除字符串1中从pos下标开始的len个长度的字符。

```
string a="hello";
```

```
a.erase(2,3);
```

```
cout<<a;
```

出现结果是：

he



substr截取子串函数

- 格式：字符串1.substr(pos,len);
- 功能：从字符串1中pos下标开始截取之后len个长度的字符得到一个新的字符串常量

```
string a="hello",b;
```

```
b=a.substr(2,3);
```

```
cout<<b;
```

出现结果是：

llo

find查找函数

- 格式：字符串1.find(字符串2,pos);//位置在后
- 功能：在字符串1中index下标所对应的元素起查找字符串2第一次出现的位置，若找到则返回第一次出现的下标；若找不到，则返回-1（原值是npos，是一个无符号整数，将返回值赋值给int型变量后正好是-1）。

```
string a="haha",b="ha";
```

```
int n=a.find(b,0);
```

```
cout<<n;
```

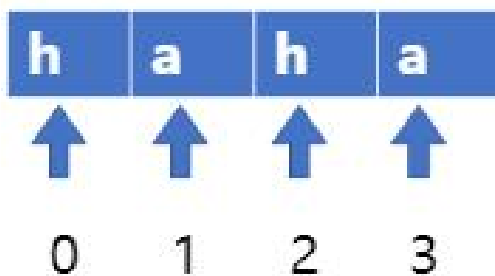
出现结果是：

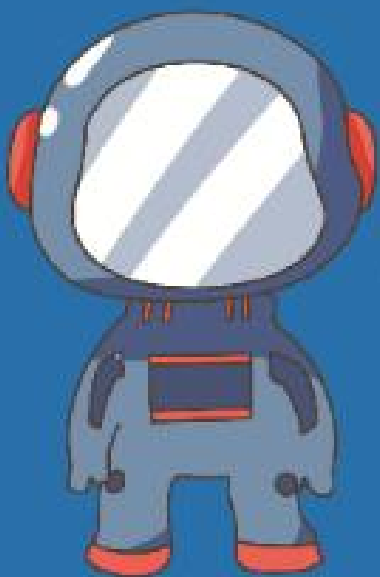
0

表示0号位置能找到

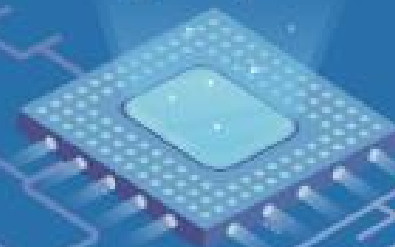
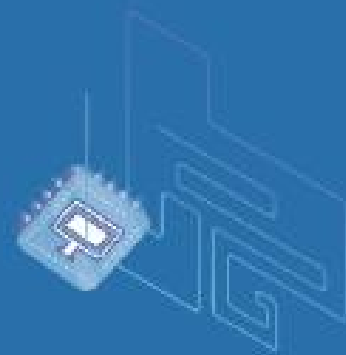
注意事项

- 以上所有函数说的pos(位置)，都是数组中对应元素之前下标的位置，就像是房子的门牌号。在找到元素之前应该有一个这样的门牌号。





课堂练习



单词填空

【描述】在一个英文句子中，有一组括号中间加一个空格“()”，需要在其中填写对应的单词。如填写单词happy，变成(happy)。现在给出未填写单词的句子和需要填写的单词，请把单词填写进去。

【输入】2行。第1行是句子，其中有且只有一个空(不是指空格blank，而是待填写的空位)，由一组括号中间加一个空格组成。第2行是要填写的单词；

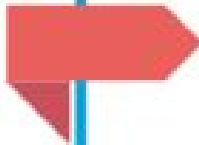
【输出】一行；填空结果；

【样例输入】

I am a ().
student

【样例输出】

I am a (student).



思路解析

- 需要使用的函数分别有查找，删除和插入函数。关键要找到在什么地方删除和插入。
- 可以考虑带括号整体删除、插入或者只进行括号中间空格的删除和插入。



参考代码

```
#include<iostream>
#include<string>
#include<cstdio>
using namespace std;
int main(){
    string a,b,c="( )";
    getline(cin,a);
    cin>>b;
    int n=a.find(c,0);//查找
    a.erase(n+1,1);//删除
    a.insert(n+1,b);//插入
    cout<<a;
    return 0;
}
```

单词替换

【描述】在一段英文语句中，发现有一个单词a使用错误，应该换成单词b。例如：we have to make homework中将make替换成do。现在需要你使用代码来实现这个功能。句子中保证要求替换的单词有且只有一个。

【输入】2行。第1行是句子，句子不含标点；第2行是两个单词，前面的是待被替换的单词，后一个是替换进去的单词；

【输出】一行；替换后的结果；

【样例输入】

I am a dog
dog student

【样例输出】

I am a student

思路解析

- 依然是查找一个字符串，但是不同的是这个字符串是单独的单词，比如查找单词do，在语句中可能出现doing,does，这些直接使用find函数查找的时候都会找到，所以要查找单独存在的单词。
- 查找单独存在的单词，可以在单词的前后加上空格，比如查找“do”变为查找“ do ”；另外，句子开头和结尾也要额外加空格，否则开头和结尾的单词找不到。
- 如果句子含有标点符号还需要根据实际情况考虑。

参考代码

```
#include<iostream>
#include<string>
using namespace std;
int main(){
    string a,b,c;
    getline(cin,a);
    cin>>b>>c;
    a=" "+a+" ";//预处理，全部加上空格，方便查找和替换；
    b=" "+b+" ";
    c=" "+c+" ";
    int n=a.find(b,0),m=b.size();
    a.erase(n,m);//删除b单词
    a.insert(n,c);//插入c单词
    a.erase(0,1);//删除前面的空格，后面的空格一般评测机会忽略比较
    cout<<a;
    return 0;
}
```

思路解析2

- 这里还提供一种字符串转为字符串数组的方法，这里因为是一个一个单词的查找和替换，如果每一个单词都是单独的字符串，那么只需要在这堆字符串（字符串数组）里面看能够找到要替换的字符串，直接跟换即可。
- 这里使用的方法是不停的寻找空格，如果不是就单词长度 t 加一，否则就把截取长度 t 的字符放入字符串数组其中一个位置，位置随之增长并且长度 t 清零。

参考代码二

```
#include<iostream>
#include<string>
using namespace std;
int main(){
    string a,b,c,s[500];
    getline(cin,a);
    cin>>b>>c;
    int n=a.size(),t=0,j=0;
    for(int i=0;i<n;i++){//在语句中查找
        if(a[i]!=' '){//不是空格就长度加1
            t++;
        }
        else{//是空格就截取放入字符串数组
            s[j++]=a.substr(i-t,t);
            t=0;//长度清零
        }
    }
```

```
        if(i==n-1){//最后一个单词的单独处理
            s[j]=a.substr(n-t,t);
        }
    }
    for(int i=0;i<=j;i++){
        if(s[i]==b){//查找有没有b这个单词
            s[i]=c;
        }
        cout<<s[i]<<" ";
    }
    return 0;
}
```

回文串

【描述】回文串是一个正读和反读都一样的字符串，比如level或者noon等等就是回文串。现在给定n个字符串，判断这些字符串是否回文串。

【输入】n+1行。第1行是n，后面接着n行字符串。

【输出】n行；判断每个字符串是否回文串，是则输出“yes”，否则输出“no”；

【样例输入】

```
3
a
level
dog
```

【样例输出】

```
yes
yes
no
```



思路解析

- 回文的标准就是字符串是否顺反一致。
- 方法一：利用对称性。看中点对称位置字符是否都相等；
- 方法二：利用概念；将原字符串前后倒转得到新串，新串和原串对比，一样就是回文。

参考代码

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    string a;
    int m;
    cin>>m;
    getline(cin,a); //还记得缓冲区bug吗?
    for(int j=1;j<=m;j++)
    {
        int flag=1;
        getline(cin,a);
        int n=a.size();
```

```
        for(int i=0;i<n/2;i++)
            if(a[i]!=a[n-1-i])
                {//前后对应查找
                    flag=0;//找到就举旗子
                    break;
                }
        if(flag==1)cout<<"yes"<<endl;
        else cout<<"no"<<endl;
    }
    return 0;
}
```

小结



1. 字符串常用函数有哪些？
2. 字符串怎么变成字符串数组？
3. 回文类型题目怎么解？



作业

- 1、字符替换 <http://noi.openjudge.cn/ch0107/08/>
- 2、字符串判等 <http://noi.openjudge.cn/ch0107/17/>
- 3、过滤多余的空格 <http://noi.openjudge.cn/ch0107/23/>
- 4、回文子串 <http://noi.openjudge.cn/ch0107/34/>