

太戈编程  
etiger.vip

# 信息学算法

# 字符串

## 综合应用

# 01串

外星人给你发送了一串**01**字符串,你想要知道字符串中究竟有多少个**1**。

注意: **01**字符串为每一个是 **0**或者**1**的字符串。

输入样例:  
**0010100**

输出样例:  
**2**

输入样例:  
**1111**

输出样例:  
**4**

虽然题目有数字概念

但是类型选择**string**  
把数字也当做符号

**string**字符串类型  
可用来输入各种类型信息

# 01串

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main(){
5      string s;
6      cin>>s;
7      int cnt=0;
8      for(int i=0;i<s.size();i++)
9          if(s[i]=='1')
10             cnt++;
11      cout<<cnt<<endl;
12      return 0;
13 }
```

cnt是count缩写  
代表计数器  
记录有几个'1'

# 错误代码 - 零分

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main(){
5      string s;
6      cin>>s;
7      int cnt; ←
8      for(int i=0;i<s.size();i++)
9          if(s[i]=='1')
10             cnt++;
11      cout<<cnt<<endl;
12      return 0;
13 }
```

运行测试  
误以为正确

高频  
错误

局部变量  
必须初始化  
不然会出现  
不确定结果

错在哪里?

# 逐位求和

输入非负整数 $n$ ,  $0 \leq n \leq 10^{100}$ 。输出 $n$ 的每一位总和。

输入样例：

1234567890123456789012345678901234567890

输出样例：

180

输入样例：

123456

输出样例：

21

输入样例：

0

输出样例：

0

虽然题目有数字概念

但是类型选择**string**  
把数字也当做符号

**string**字符串类型  
可用来输入各种类型信息

# 逐位求和

```
1 #include<iostream>
2 #include<string>
3 using namespace std;
4 int main(){
5     string s;
6     getline(cin,s);
7     int ans=0; ← 易错点
8     for(int i=0;i<s.size();i++)
9         ans+=s[i]-'0';
10    cout<<ans<<endl;
11    return 0;
12 }
```

'9'-'0'是整数9  
'8'-'0'是整数8  
'1'-'0'是整数1

'B'-'A'是整数1  
'C'-'A'是整数2  
'z'-'a'是整数25

两个字符相减  
得到整数答案  
基于ASCII码



# 身份证检验码

身份证共18位，最后一位是检验码。计算规则如下：

- 1.前17位数分别乘不同系数:7 9 10 5 8 4 2 1 6 3 7 9 10 5 8 4 2
- 2.将这17位数字和系数相乘的结果相加
- 3.总和除以11取余数
- 4.余数只可能有0 1 2 3 4 5 6 7 8 9 10这11个数字。其分别对应的最后一位身份证的号码为1 0 X 9 8 7 6 5 4 3 2

例如：身份证号码是34052419800101001X

计算 $3*7+4*9+0*10+5*5+2*8...+1*2$ ，前17位的乘积和是189。然后189除以11余2。余数2对应的数字是X。所以这个身份证检验正确。





# 身份证检验码

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	--

乘法系数

a[i]	7	9	10	5	8	4	2	1	6	3	7	9	10	5	8	4	2	
------	---	---	----	---	---	---	---	---	---	---	---	---	----	---	---	---	---	--

身份证s

s[i]	'3'	'1'	'0'	'1'	'0'	'1'	'2'	'0'	'0'	'8'	'0'	'1'	'0'	'1'	'3'	'1'	'1'	'?'
------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

C++代码

```
int a[20]={7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2};
```

前17位成绩求和:

$$\begin{aligned} &3*7+1*9+0*10+1*5+0*8+1*4+2*2+0*1+0*6 \\ &+8*3+0*7+1*9+0*10+1*5+3*8+1*4+1*2 = 111 \end{aligned}$$

请手算  
结果

111%11结果1

# 身份证检验码

i	0	1	2	3	4	5	6	7	8	9	10
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
code[i]	'1'	'0'	'X'	'9'	'8'	'7'	'6'	'5'	'4'	'3'	'2'

## C++代码

```
char code[20]={'1','0','X','9','8','7','6','5','4','3','2'};
```

前17位成绩求和：

$$3*7+1*9+0*10+1*5+0*8+1*4+2*2+0*1+0*6 \\ +8*3+0*7+1*9+0*10+1*5+3*8+1*4+1*2=111$$

111%11结果1

1对应的检验码为code[1]也就是'0'

a[i]代表i号位置  
对应的系数

code[i]代表余数  
i对应的检验码

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int a[20]={7,9,10,5,8,4,2,1,6,3,7,9,10,5,8,4,2};
5  char code[20]={'1','0','X','9','8','7','6','5','4','3','2'};
6  int main(){
7      string s;
8      cin>>s;
9      int sum=0;
10     for(int i=0;i<=16;i++)
11         sum+=(s[i]-'0')*a[i];
12     sum%=11;
13     cout<<code[sum]<<endl;
14     return 0;
15 }
```

翻译每一行  
写注释

# 字符串

## 成员函数

求大小	<code>.size()</code>
取子串	<code>.substr()</code>
删除	<code>.erase()</code>
查找	<code>.find()</code>
插入	<code>.insert()</code>
替换	删除+插入

# 函数复习

纸和笔  
完成以下  
C++语句

```
string s="abcdefgh";
```

删除s的首个字符

```
s.erase(0,1);
```

取出s的子串:从2号开始取3个字符,赋值给字符串t

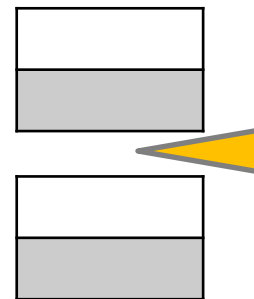
```
string t=s.substr(2,3);
```

在s中寻找"cd"的第一个开始位置,赋值给整数p

```
int p=s.find("cd");
```

字符串插入

`.insert()`



# .insert() 插入字符串

```
string s="I am lucky";
```

```
s.insert(5,"very ");
```

```
cout<<s<<endl;           I am very lucky
```

```
s.insert(10,"un");
```

```
cout<<s<<endl;           I am very unlucky
```

```
s.insert(5,"not ");
```

```
cout<<s<<endl;           I am not very unlucky
```

`s.insert(p,t)`函数的功能：  
在字符串s中p号位置插入字符串t



# 阅读程序猜结果

```
string s="cici";  
s.insert(2,"da");  
cout<<s<<endl;  
  
s.insert(0,"dong");  
cout<<s<<endl;
```

注意: `.insert()` 函数会改变原字符串内容

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main(){
5      string s="water";
6      s+="melon";
7      cout<<s<<endl;
8      cout<<s.length()<<endl;
9      cout<<s.substr(0,5)<<endl;
10     cout<<s.substr(5)<<endl;
11     s.erase(0,5);
12     cout<<s<<endl;
13     s.insert(0,"hami ");
14     cout<<s<<endl;
15     return 0;
16 }
```

轮流口头  
翻译每一行

# 替换1

输入一行多个单词，由空格隔开。请把第一个单词替换成\*\*\*

输入样例：

Mike is the real murderer

输出样例：

\*\*\* is the real murderer

输入样例：

Wangxiaoming didn't pass the exam

输出样例：

\*\*\* didn't pass the exam

输入一整行字符串

逐位查看每一个字母

如果发现空格，  
就能定位第一个单词

将该单词替换为\*\*\*

替换:先删除再插入

# 替换1

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main(){
5      string s;
6      getline(cin,s);
7      int p;
8      for(p=0;p<s.size();p++)
9          if(s[p]==' ') break;
10     s.erase(0,p);
11     s.insert(0,"***");
12     cout<<s<<endl;
13     return 0;
14 }
```

输入一整行字符串

逐位查看每一个字母

如果发现空格  
跳出循环

先删除  
再插入

# 替换2

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  int main(){
5      string s;
6      getline(cin,s);
7      int a=s.find(" ");
8      int b=s.find(" ",a+1);
9      s.erase(a+1,b-a-1);
10     s.insert(a+1,"***");
11     cout<<s<<endl;
12     return 0;
13 }
```

输入一整行字符串

找到第一个空格  
和第二个空格

先删除  
再插入

# 替换2

输入一行多个单词，由空格隔开。请把**第二个**单词替换成\*\*\*

输入样例：

Mike Chen is the real murderer

输出样例：

Mike \*\*\* is the real murderer

输入样例：

Wang Da Ming didn't pass the exam

输出样例：

Wang \*\*\* Ming didn't pass the exam

输入一整行字符串

找到第一个空格和第二个空格

就能定位第二个单词

将该单词替换为\*\*\*

替换:先**删除**再**插入**

# 现场挑战

## 143



编号

0	1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	---	----	----	----

ISBN

0	-	6	7	0	-	8	2	1	6	2	-	4
---	---	---	---	---	---	---	---	---	---	---	---	---

6

```
string s;
```

7

```
cin>>s;
```

8

```
int c=1, sum=0;
```

9

```
for(int i=0; i<s.size()-2; i++)
```

10

```
    if(i!=1&&i!=5) {
```

11

```
        sum+=c*(s[i]-'0');
```

12

13

```
    }
```

14

```
int id=sum%11;
```

c表示什么含义

sum表示什么含义

编号

0	1	2	3	4	5	6	7	8	9	10	11	12
---	---	---	---	---	---	---	---	---	---	----	----	----

ISBN

0	-	6	7	0	-	8	2	1	6	2	-	4
---	---	---	---	---	---	---	---	---	---	---	---	---

```
14 int id=sum%11;
15 char code='X';
16 if(id<10)
17     code='0'+id;
18 if(s[12]==code)
19     cout<<"Right";
20 else
21     cout<<[ ]<<endl;
```

# 太戈编程

104

985

143

拓展题

142,800