



# [6] 字符串与文件操作

深入浅出程序设计竞赛  
第 1 部分 – 语言入门  
V 2021-02

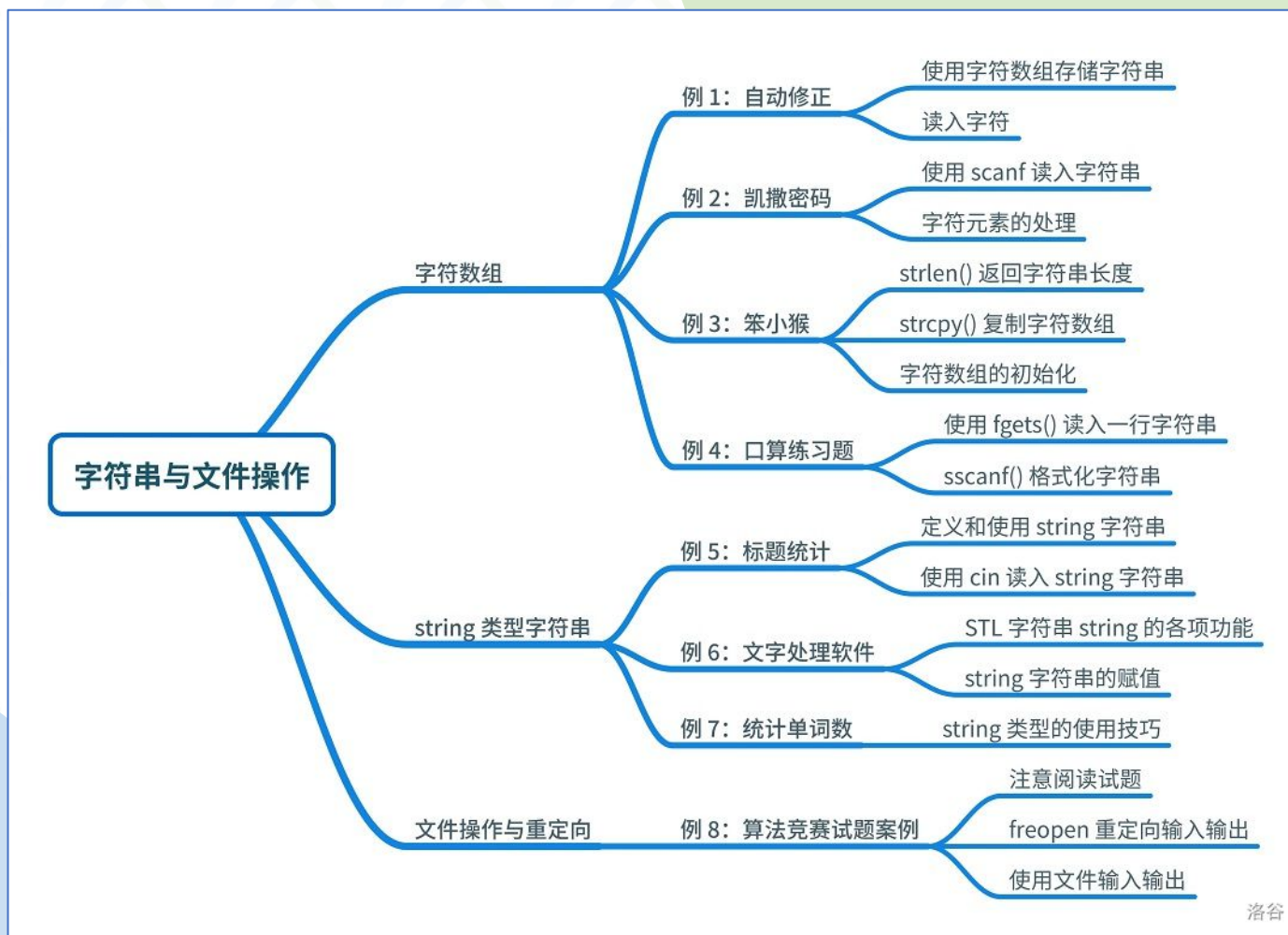
## 版权声明

本课件为《深入浅出程序设计竞赛 - 基础篇》的配套课件，版权归 洛谷 所有。所有个人或者机构均可免费使用本课件，亦可免费传播，但不可付费交易本系列课件。

若引用本课件的内容，或者进行二次创作，请标明本课件的出处。

- 其它《深基》配套资源、购买本书等请参阅：  
<https://www.luogu.com.cn/blog/kkksc03/IPC-resources>
- 如果课件有任何错误，请在这里反馈  
<https://www.luogu.com.cn/discuss/show/296741>

# 本章知识导图



## 第6章 字符串与文件操作

---

字符数组

string 类型字符串

文件操作与重定向

课后习题与实验

# 字符数组

出了计算数字，计算机还可以处理文章和句子。那计算机是如何存储和处理这些文字数据呢？

请翻至课本 P82



# 字符数组

字符数组实质上 and 整数数组没什么区别，只是数组中的每一个元素都是一个字符（实际上，都是存成对应 ASCII 的数字）。

将这些字符存储下来，便组成了一串字符，可以进行进一步操作。

数字	字符	数字	字符	数字	字符	数字	字符	数字	字符	数字	字符	数字	字符	数字	字符
32	[空格]	44	,	56	8	68	D	80	P	92	\	104	h	116	t
33	!	45	-	57	9	69	E	81	Q	93	]	105	i	117	u
34	"	46	.	58	:	70	F	82	R	94	^	106	j	118	v
35	#	47	/	59	;	71	G	83	S	95	_	107	k	119	w
36	\$	48	0	60	<	72	H	84	T	96	`	108	l	120	x
37	%	49	1	61	=	73	I	85	U	97	a	109	m	121	y
38	&	50	2	62	>	74	J	86	V	98	b	110	n	122	z
39	'	51	3	63	?	75	K	87	W	99	c	111	o	123	{
40	(	52	4	64	@	76	L	88	X	100	d	112	p	124	
41	)	53	5	65	A	77	M	89	Y	101	E	113	q	125	}
42	*	54	6	66	B	78	N	90	Z	102	F	114	r	126	~
43	+	55	7	67	C	79	O	91	[	103	g	115	s		

# 自动修正

---

## 例 6.1 (洛谷 P5733)

大家都知道一些办公软件有自动将字母转换为大写的功能。

输入一个长度不超过 100 且不包括空格的字符串。

要求将该字符串中的所有小写字母变成大写字母并输出。

例如输入 `Luogu4!`，输出 `LUOGU4!`。

# 自动修正

单个字符使用 `char` 类型存储，回想上一章刚刚介绍过的数组，存储一排字符是可以使用数组。

定义一个数组，每一个元素都是字符类型，该数组就可存字符串。

定义和读入：

```
char s[数组长度];  
scanf("%s", s); 或者 cin >> s;
```

只能读入到空格  
或者换行符为止

```
#include <iostream>  
#include <cstdio>  
using namespace std;  
int main() {  
    char s[110];  
    scanf("%s", s); // 读入这个字符串，还可以 cin>>s;  
    for (int i = 0 ; s[i] != '\0' ; i++)  
        if ('a' <= s[i] && s[i] <= 'z')  
            // 如果这个字符在'a'到'z'中间，说明是小写字母  
            s[i] -= 'a' - 'A'; // 变成对应的大写字母，减去偏移量  
    printf("%s\n", s); // 输出，还可 cout<<s<<endl;  
    return 0;  
}
```



## 自动修正

数组 `s` 中每一项储存不超过 127 的整数，代表对应 ASCII 编码。

这个字符串实际上在字符数组中是这么被储存的：

s[0]	s[1]	s[2]	s[3]	s[4]	a[5]	s[6]	s[7]
76' L'	117' u'	111' o'	103' g'	117' u'	52' 4'	33' '!	0' \0'

虽然只有 7 个字符，但这个字符数组却占了 8 位 (`s[0]` 到 `s[7]`)

字符串结束后，还需要有一个特殊的“结束标记字符”——

`'\0'`，其对应的数字就是 0。输出时碰到这个标记就停止输出。

其他的一些特殊字符：

'\0'	'\n'	'\"'	'\\'
结束标记	换行	单引号	斜杠

# 自动修正

另解：每次读入一个字符，判断是否需要处理，再直接输出字符。

使用 `getchar()` 函数获取输入数据中一个字符

使用 `putchar()` 函数则是输出一个字符。

```
#include <iostream>
#include <cstdio>
using namespace std;
int main() {
    char s;
    while (1) {
        s = getchar(); // 每次调用getchar()函数，读入一个字符
        if (s == EOF) break;
        if ('a' <= s && s <= 'z') // 如果这个字符是小写字母
            s += 'A' - 'a'; // 把它转换成大写字母，这么写和上面是一样的
        putchar(s); // 调用putchar()函数，输出一个字符
    }
    return 0;
}
```

输入完字符串后，按 `CTRL+Z` 后回车。  
如果文件被读完了，那么 `getchar()` 函数会返回 `EOF` 提示程序输入已经完毕

# 凯撒密码

## 例 6.2 (洛谷 P1914)

凯撒密码是由原文字符串（由不超过 50 个小写字母组成）中每个字母向后移动  $n$  位形成的。z 的下一个字母是 a，如此循环。

给出  $n$  和移动前的原文字符串，请你求出密码。

1  
qwe

rxfr

# 凯撒密码

读入字符串，然后将每个字符处理后输出。直接加上偏移量？

要求 'z' 的下一个是 'a'，如此循环，因此不能直接加上偏移量。

计算这个小写字母是字母表第几个： $s[i] - 'a'$

目标字母位置需加上  $n$ ： $s[i] - 'a' + n$

这个数可能超过 25，所以需要对 26 取模： $(s[i] - 'a' + n) \% 26$

最后还要加回 'a' 变成对应的字母 ASCII。

```
int n; char s[60];
scanf("%d %s", &n, s); //读入字符串
for (int i = 0; s[i]; i++)
    putchar((s[i] - 'a' + n) % 26 + 'a'); // 计算偏移量并还原
```



# 笨小猴

例 6.3 (洛谷 P1125, NOIP2008 提高组)

给出一个单词 (由不超过 100 个小写字母组成)

假设  $\text{maxn}$  是单词中出现次数最多的字母的出现次数,  $\text{minn}$  是单词中出现次数最少的字母的出现次数。

如果  $\text{maxn} - \text{minn}$  是一个质数, 那么笨小猴就认为这是个 Lucky Word, 输出 Lucky Word, 然后在第二行输出  $\text{maxn} - \text{minn}$  的值; 否则输出 No Answer, 第二行输出 0。

error

Lucky Word  
2

olympic

No Answer  
0



# 笨小猴

读入每个字符串之后，用数组记录 ‘a’ 到 ‘z’ 中每个字母出现次数

用 “打擂台” 思路寻找出现次数最多和最少字母，判断是否相等

```
#include <iostream>
#include <cstring>
using namespace std;
int main() {
    char a[110];
    int ans[26] = {0};
    int l, mmax, mmin, delta;
    scanf ("%s", a);
    l = strlen (a);
    for (int i = 0; i < l; i++)
        ans[a[i] - 'a']++;
    mmax = 0; mmin = 10000;
    for (int i = 0; i < 26; i++) {
        if (ans[i] > mmax) mmax = ans[i];
        if (ans[i] != 0 && ans[i] < mmin)
            mmin = ans[i];
    }
}
```

```
delta = mmax - mmin
if (delta == 0 || delta == 1) {
    printf ("No Answer\n0\n");
    return 0;
}
for (int h = 2; h*h <= delta; h++)
    if (delta % h == 0) {
        printf ("No Answer\n0\n");
        return 0;
    }
puts ("Lucky Word\n");
printf ("%d\n", mmax-mmin);
return 0;
}
```

//两句话可以写在一起

```
printf ("Lucky Word\n%d\n", mmax-mmin);
```

# 字符数组相关函数

注意使用到了 `cstring` 头文件，其中一些的用法如下：

<code>strlen(s)</code>	获得字符数组的 <b>长度</b> ，数到 ' \0'
<code>strcpy(a,b)</code>	将 b 字符数组的数据 <b>复制</b> 到 a
<code>strcmp(a,b)</code>	比较两个字符串，a>b 则返回 1
<code>char a[100] ; strcpy(a, "hello");</code>	给字符数组 <b>赋值常量</b>

**不能**直接给字符数组赋值，除非在定义时初始化。

```
char a[100]="Luogu!"
```

从输入中读入完整一行（含空格）可这么写。**不要用** `gets` ！

```
fgets(s, sizeof(s), stdin);
```

因为可能造成缓冲区溢出的安全问题，C++新标准不再建议使用 `gets`。

# string 类型字符串

使用 C 语言风格的字符数组有诸多不便，于是再在 C++ 中提供了一些更好的工具，将很多有用的功能进行了封装，开箱即用。

请翻至课本 P87

# 标准模板库

---

字符数组的局限：不能弹性变化长度、不能直接赋值或者复制，也有数组越界的风险。

可使用标准模板库（Standard Template Library, STL）

包括各类容器（比如队列、栈等）、算法（比如排序）等

STL 中的字符串（string）容器可以帮我们带来很多便利

# 标题统计

例 6.5 (洛谷 P5015, NOIP2018 普及组)

给定一个字符串，请问有多少个字符？

注意：标题中可能包含大、小写英文字母、数字字符、空格和换行符，且字符串中的字符和空格数总和不超过 5。统计标题字符数时，空格和换行符不计算在内。

234

3

Ca 45

4

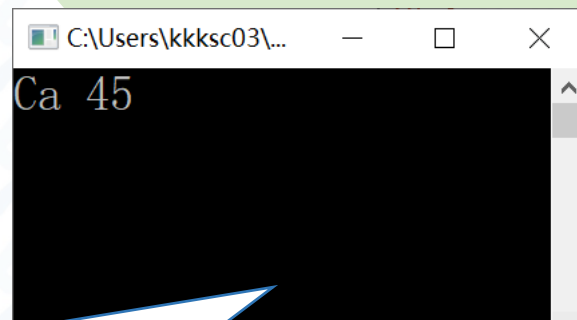


# 标题统计

读入字符串时读到分隔符（空格换行）时停止，故一直读入字符串，每读一个字符串，就把其长度加入答案。

使用了 `string` 头文件并定义 `s` 字符串。

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string s;
    int ans = 0;
    while (cin >> s)
        ans += s.length();
    cout << ans << endl;
    return 0;
}
```



没有反应？按 `CTRL+Z` 后回车。  
如果文件被读完了，那么 `cin >> s`  
返回 `false` 提示程序输入已经完毕

用起来很方便，`s.length()` 可返回 `s` 的长度，`s[0]` 代表最开始字符

# 文字处理软件

## 例 6.6 (洛谷 P5734)

输入一个字符串文档，开头是第 0 个字符。需支持以下操作：

- 1 str：在文档后面插入字符串 str，并输出文档的字符串。
- 2 a b：只保留从第 a 个字符起 b 个字符，并输出文档。
- 3 a str：在文档中第 a 个字符前面插入字符串 str，并输出。
- 4 str：查字符串 str 在文档中最先位置并输出；找不到输出 -1。

初始文档和每次操作的 str 无空格或换行。最多  $q(q \leq 100)$  次操作。

```
4
ILove
1 Luogu
2 5 5
3 3 guGugu
4 gu
```

```
ILoveLuogu
Luogu
LuoguGugugu
3
```

# 文字处理软件

使用 `string` 头文件来操作字符串，其中一些的用法如下：

功能	说明
<code>string s;</code>	定义一个名字为 <code>s</code> 的字符串变量
<code>s+=str</code> 或 <code>s.append(str)</code>	在字符串 <code>s</code> 后面拼接字符串 <code>str</code>
<code>s&lt;str</code>	比较字符串 <code>s</code> 的是否在字符串 <code>str</code> 的字典序之前
<code>s.size()</code> 或 <code>s.length()</code>	得到字符串 <code>s</code> 的长度
<code>s.substr(pos,len)</code>	截取字符串 <code>s</code> ，从第 <code>pos</code> 个位置开始 <code>len</code> 个字符，并返回这个字符串
<code>s.insert(pos, str)</code>	字符串 <code>s</code> 的第 <code>pos</code> 个字符之前，插入字符串 <code>str</code> ，并返回这个字符串
<code>s.find(str, [pos])</code>	字符串 <code>s</code> 中从第 <code>pos</code> 个字符开始寻找 <code>str</code> ，并返回位置，如果找不到返回 <code>string::npos</code> 。 <code>pos</code> 可以省略，默认值是 0

将其强制转换为 `int` 类型后才是 -1

# 文字处理软件

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    int n, opt, l, r;
    string s, a;
    cin >> n;
    cin >> s;
    while (n--) {

        //见右边

    }
    return 0;
}
```

string 类型就要简单很多  
可直接赋值常量，  
也可以相互赋值。

```
cin >> opt;
if (opt == 1) {
    cin >> a;
    s.append(a);
    //使用append函数，将a字符串加在s字符串后面
    cout << s << endl;
} else if (opt == 2) {
    cin >> l >> r;
    s = s.substr(l, r);
    //使用substr函数，提取出s从l起的r个字符
    cout << s << endl;
} else if (opt == 3) {
    cin >> l >> a;
    s.insert(l, a);
    //使用insert函数，将a字符串插入到l位置
    cout << s << endl;
} else {
    cin >> a;
    cout << (int)s.find(a) << endl;
    //用find函数，输出a串在s串中第一次出现的位置
}
```

# 字符数组和字符串的异同

- 字符数组：C 语言就存在，比较原始；
- string：在 C++ 中可以使用字符数组的“进化版本”。

string 的变量名在很多情况下可当做字符数组的变量名，用于 sscanf、sprintf 等地方。相关内容见 P87。

string 和字符数组也是可以相互转换的，见下方例子：

```
// string转字符数组
char arr[10];
string s = "LUOGU";
int len = s.copy(arr, 9);
// 最多允许复制9个字符，否则就越界了
arr[len] = '\0'; // 在末尾增加结束标记
// 或者
char arr[10];
string s = "LUOGU";
strcpy(arr, s.c_str());
// strncpy(arr, s.c_str(), 10);
```

```
// 字符数组转string就更简单了
char arr[10];
strcpy(arr, "LUOGU");
string s;
s = arr;
```

提示：读入一整行的 string 字符串  
getline(cin, 字符串名称)



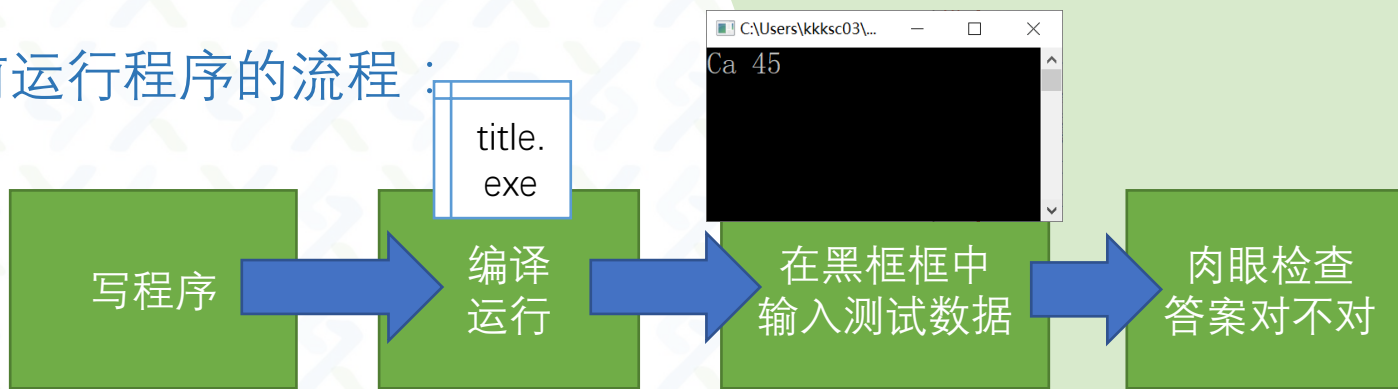
# 文件操作与重定向

许多程序设计竞赛（比如 NOI 系列比赛）要求使用文件输入输出。如果没有按照要求正确地使用文件输入输出，即使算法完全正确，也不能获得分数。

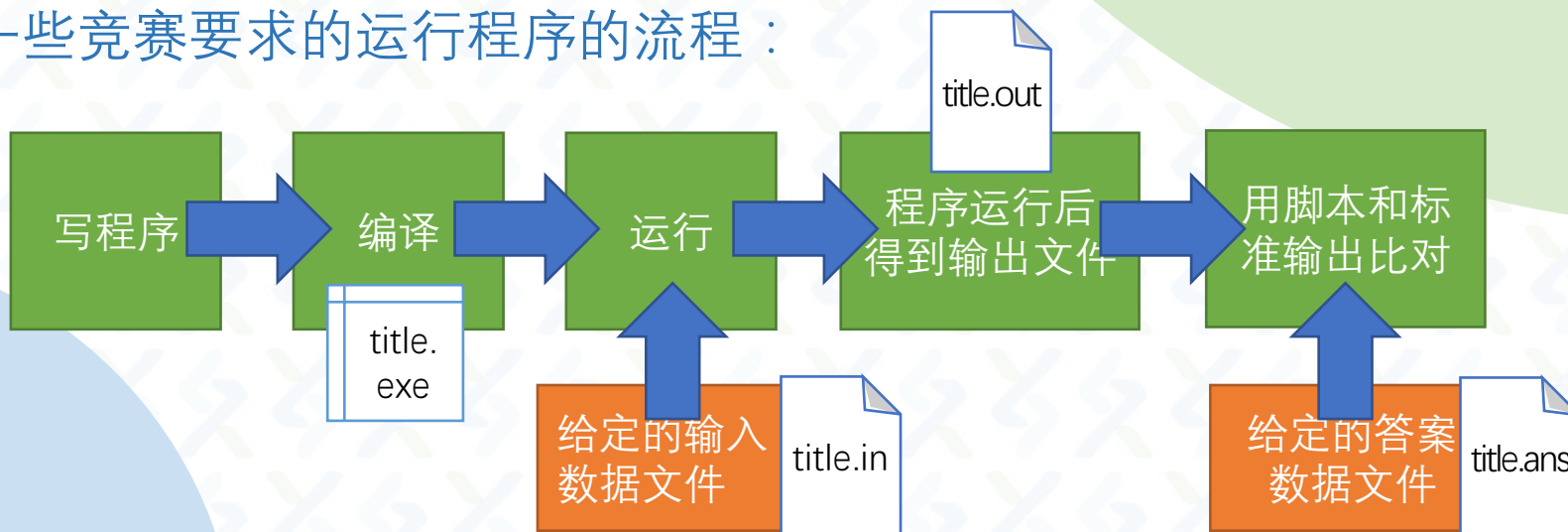
请翻至课本 P91

# 文件操作

之前运行程序的流程：



一些竞赛要求的运行程序的流程：



# 文件操作

---

## 标准输入输出：

弹出一个窗口，手动输入内容，程序运行后在同一窗口输出结果。  
包括洛谷在内，多数 Online Judge 都这样对程序进行评判。

## 文件输入输出：

这种方式可以将硬盘上的输入文件“喂给”程序；  
然后程序运算后“吐出”另外一个输出文件。

# 标题统计

---

正式的试题包括 文件名要求、题目描述、输入/输出格式、样例（输入/输出样例和样例解释）和数据范围等。

必须要认真阅读每一项信息，理解题目的要求。

题目本身很简单，但如果未按要求正确使用文件输入输出，即使算法完全正确，也不能获得分数。

接下来请大家阅读 NOIP 2018 普及组复赛 的试题，并且特别注意画圈的部分。

## CCF 全国信息学奥林匹克联赛（NOIP2018）复赛

## 普及组

（请选手务必仔细阅读本页内容）

## 一. 题目概况

中文题目名称	标题统计	龙虎斗	摆渡车	对称二叉树
英文题目与子目录名	title	fight	bus	tree
可执行文件名	title	fight	bus	tree
输入文件名	title.in	fight.in	bus.in	tree.in
输出文件名	title.out	fight.out	bus.out	tree.out
每个测试点时限	1 秒	1 秒	2 秒	1 秒
测试点数目	20	25	20	25
每个测试点分值	5	4	5	4
附加样例文件	有	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）			
题目类型	传统	传统	传统	传统
运行内存上限	256M	256M	256M	256M



## 1. 标题统计

`(title.cpp/c/pas)`

### 【问题描述】

凯凯刚写了一篇美妙的作文，请问这篇作文的标题中有多少个字符？

注意：标题中可能包含大、小写英文字母、数字字符、空格和换行符。统计标题字符数时，空格和换行符不计算在内。

### 【输入格式】

输入文件名为 `title.in`。

输入文件只有一行，一个字符串 `s`。

### 【输出格式】

输出文件名为 `title.out`。

输出文件只有一行，包含一个整数，即作文标题的字符数（不含空格和换行符）。

### 【输入输出样例 1】

<code>title.in</code>	<code>title.out</code>
234	3

见选手目录下的 `title/title1.in` 和 `title/title1.ans`。

### 【输入输出样例 1 说明】

标题中共有 3 个字符，这 3 个字符都是数字字符。

# 标题统计

## 【输入输出样例 2】

title.in	title.out
Ca 45	4

见选手目录下的 title/title2.in 和 title/title2.ans。

## 【输入输出样例 2 说明】

标题中共有 5 个字符, 包括 1 个大写英文字母, 1 个小写英文字母和 2 个数字字符, 还有 1 个空格。由于空格不计入结果中, 故标题的有效字符数为 4 个。

## 【数据规模与约定】

规定  $|s|$  表示字符串  $s$  的长度 (即字符串中的字符和空格数)。

对于 40% 的数据,  $1 \leq |s| \leq 5$ , 保证输入为数字字符及行末换行符。

对于 80% 的数据,  $1 \leq |s| \leq 5$ , 输入只可能包含大、小写英文字母、数字字符及行末换行符。

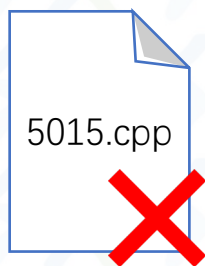
对于 100% 的数据,  $1 \leq |s| \leq 5$ , 输入可能包含大、小写英文字母、数字字符、空格和行末换行符。

# 标题统计

刚刚分析了第一题“标题统计”的做法，写出了一段程序代码。

但是选手在赛场上不能直接提交写完的程序，必须要按照题目的要求提交代码，特别注意以下信息：

- 提交源程序文件名，对于 C++：title.cpp
- 输入文件名：title.in
- 输出文件名：title.out



Windows 下文件名  
不区分大小写  
但 Linux 下区分

# 标题统计

## 原本的程序

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string s;
    int ans = 0;
    while (cin >> s)
        ans += s.length();
    cout << ans << endl;
    return 0;
}
```

## 需要提交的程序

```
#include <iostream>
#include <cstdio>
#include <string>
using namespace std;
int main() {
    freopen("title.in", "r", stdin);
    freopen("title.out", "w", stdout);
    string s;
    int ans = 0;
    while (cin >> s)
        ans += s.length();
    cout << ans << endl;
    return 0;
}
```

# 标题统计

在主程序的开头多出了两句话，同时还使用了 `cstdio` 头文件，这就是 重定向输入输出到文件 的方式，其一般形式为：

```
freopen("输入文件名", "r", stdin);  
freopen("输出文件名", "w", stdout);
```

在 `title.cpp` 相同的文件夹里新建一个文件 `title.in`，使用记事本打开这个文件，把样例输入 `234` 复制进去并保存。

重新运行程序，在文件夹里出现了一个 `title.out` 的文件。



使用记事本打开这个文件，这就是这个程序的输出。



# 文件操作的小提示

文件输入输出的好处：不需每次运行程序都要用键盘敲一遍输入。在记事本中写好输入文件并保存好，直接就可以运行程序了，节约时间。

在本地使用文件输入输出，在线提交 OJ 使用标准输入输出：

```
#ifndef ONLINE_JUDGE
    freopen("title.in", "r", stdin);
    freopen("title.out", "w", stdout);
#endif
```

# 课后习题与实验

学而时习之，不亦说乎。学而不思则罔，思而不学则殆。——孔子

请翻至课本 P95

# 总结

---

## 字符数组

就是数组中存储字符的数组，知道如何输入输出  
对数组中的字符进行处理、修改

## STL 中的字符串 string

STL 提供了很多封装好的工具可以使用  
功能强大，使字符串处理的问题简单很多

## 文件输入输出

在一些竞赛中必须使用文件输入输出  
相比于标准输入输出，在本地调试可能更省时间

## 课后拓展

### 习题 6.1 手机 (洛谷 P1765)

一般的手机键盘如右图：

要按出英文字母就要按数字键多下。

例如要按出 x 就得按 9 两下：第一下会出 w，而第二下把 w 变成 x。

0 键按一下会出一个空格。

你的任务是读取若干句只包含英文小写字母和空格的句子，求出要在手机上打出这个句子至少需要按多少下键盘。

1	2 abc	3 def
4 ghi	5 jkl	6 mno
7 pqrs	8 tuv	9 wxyz
*	0	#



## 课后拓展

### 习题 6.2 : honoka 的键盘 (洛谷 P3741)

用一个只有 V 和 K 两个键的键盘打出了个只有 V 或 K 的字符串。

当这个字符串里含有 "VK" 时, 就特别喜欢这个字符串。

改变至多一个字符 (或者不做任何改变) 来最大化这个字符串内 "VK" 出现的次数。只有当 "V" 和 "K" 正好相邻时, 则认为出现了 "VK"。

### 习题 6.3 : 单词覆盖还原 (洛谷 P1321)

一个长度为  $l$  ( $3 \leq l \leq 255$ ) 的字符串中反复贴有 boy 和 girl 两单词, 后贴上的可能覆盖已贴上的单词 (没有被覆盖的用句点表示), 最终每个单词至少有一个字符没有被覆盖。

问贴有几个 boy 几个 girl ?



## 课后拓展

### 习题 6.4：数字反转 - 升级版（洛谷 P1553）

给定一个数，请将该数各个位上数字反转得到一个新数。

整数反转是将所有数位对调；

小数反转是把整数部分的数反转，再将小数部分的数反转；

分数反转是把分母的数反转，再把分子的数反转；

百分数的分子一定是整数，百分数只改变数字部分。

5087462

600.084

700/27

8670%

2647805

6.48

7/72

768%

## 课后拓展

习题 6.6：你的飞碟在这儿（洛谷 P1200, USACO Training）

输入两行，每行一个长度 1 到 6 的字符串，对应一个数字：

将这个字符串中每个字母对应的数字（A 对应 1，B 对应 2……Z 对应 26）相乘，然后对 47 取模。

如果这两个字符串对应的数字相同，输出 GO，否则输出 STAY。

习题 6.7：语句解析（洛谷 P1597）

一串长度不超过 255 的 PASCAL 语言代码，只有 a,b,c 3 个变量。只有赋值语句，赋值只能是一个一位的数字或一个变量

每条赋值语句的格式是 [变量]:=[变量或一位整数];。

未赋值的变量值为 0。输出 a、b、c 的值。

```
a:=3;b:=4;c:=5;
```

```
3 4 5
```

## 参考阅读材料

---

以下内容限于课件篇幅未能详细阐述。如果学有余力，可自行翻阅课本作为扩展学习。

- P86 例 6.4：使用 `fgets()` 读入字符串，用 `sscanf()` 格式化
- P90 例 6.7：string 的使用技巧
- 习题 6.5、6.8。