

## 字符数组、string

C 语言不存在字符串，使用字符数组来保存字符。

c++ 可以使用字符数组和字符串。

char s 字符类型，只能输入单个字符

char s[200] 字符数组，cin>>s 只能保存连续的单词，空格不能保存

gets() 可以保存带空格的数组（gets 已经弃用，不再使用，仅做解释）

fgets(a, 5, stdin) 可以保存带空格的数组

cin.getline(s, 200) 可以保存带空格的数组

getline 所属 iostream 类，作用是读取一整行，通过换行符确定读取结束，可以读取空格。

getchar 逐个读入，putchar 逐个输出。

cin、scanf、getchar 以空格作为结束，get、fgets、cin.getline 以回车（换行，\n）作为结束。

字符数组以“\0”结尾，例如，abc，长度为 4，自动加 \0。

字符数组长度固定，string 长度不固定。

字符数组不能直接进行比较运算，string 可以。字符数组使用 =、>、<，相当于比较字符数组第一个元素的地址。

string 是 c++ 的 stl（标准模板库）提供的字符串类，用于处理字符串相关的问题。

size() 用于测量字符串长度，不能测量字符数组。

l=strlen(s), len=strlen(s)，与 #include<cstring> 一起使用，用于测量字符数组长度

sizeof() 用于测量数组占用空间大小。

字典码：从字符串第一个元素比较，如果有结果，比较结束。

A--65, a--97, 0--48。

# 字符、ASCII 表

ASCII表																									
( American Standard Code for Information Interchange 美国标准信息交换代码 )																									
高四位	ASCII控制字符															ASCII打印字符									
	0000					0001					0010	0011	0100	0101	0110	0111									
	0					1					2	3	4	5	6		7								
低四位	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	Ctrl	代码	转义字符	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	Ctrl
0000	0		^@	NUL	\0	空字符	16	▶	^P	DLE	数据链路转义	32		48	0	64	@	80	P	96	`	112	p		
0001	1	☺	^A	SOH		标题开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q		
0010	2	☹	^B	STX		正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r		
0011	3	♥	^C	ETX		正文结束	19	!!	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s		
0100	4	♦	^D	EOT		传输结束	20	¶	^T	DC4	设备控制 4	36	\$	52	4	68	D	84	T	100	d	116	t		
0101	5	♣	^E	ENQ		查询	21	§	^U	NAK	否定应答	37	%	53	5	69	E	85	U	101	e	117	u		
0110	6	♠	^F	ACK		肯定应答	22	—	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v		
0111	7	•	^G	BEL	\a	响铃	23	↕	^W	ETB	传输块结束	39	'	55	7	71	G	87	W	103	g	119	w		
1000	8	▣	^H	BS	\b	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x		
1001	9	○	^I	HT	\t	横向制表	25	↓	^Y	EM	介质结束	41	)	57	9	73	I	89	Y	105	i	121	y		
1010	A	◻	^J	LF	\n	换行	26	→	^Z	SUB	替代	42	*	58	:	74	J	90	Z	106	j	122	z		
1011	B	♂	^K	VT	\v	纵向制表	27	←	^[	ESC	溢出	43	+	59	;	75	K	91	[	107	k	123	{		
1100	C	♀	^L	FF	\f	换页	28	└	^\	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124			
1101	D	♪	^M	CR	\r	回车	29	↔	^_	GS	组分隔符	45	-	61	=	77	M	93	]	109	m	125	}		
1110	E	🎵	^N	SO		移出	30	▲	^^	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~		
1111	F	☼	^O	SI		移入	31	▼	^~	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	△	^Backspace 代码: DEL	

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。 2013/08/08

0	16	▶	32	48	0	64	@	80	P	96	`	112	p	128	Ç	144	É	160	á	176	☼	192	L	208	ll	224	α	240	≡		
1	☺	17	◀	33	!	49	1	65	A	81	Q	97	a	113	q	129	ü	145	æ	161	í	177	≡	193	└	209	≡	225	ß	241	±
2	☹	18	↕	34	"	50	2	66	B	82	R	98	b	114	r	130	é	146	Æ	162	ó	178	☼	194	└	210	ll	226	Γ	242	≥
3	♥	19	!!	35	#	51	3	67	C	83	S	99	c	115	s	131	â	147	ô	163	ú	179		195	└	211	ll	227	π	243	≤
4	♦	20	¶	36	\$	52	4	68	D	84	T	100	d	116	t	132	ä	148	ö	164	ñ	180	└	196	—	212	ll	228	Σ	244	∫
5	♣	21	§	37	%	53	5	69	E	85	U	101	e	117	u	133	à	149	ò	165	Ñ	181	≡	197	└	213	ll	229	σ	245	∫
6	♠	22	—	38	&	54	6	70	F	86	V	102	f	118	v	134	â	150	û	166	ª	182	ll	198	ll	214	ll	230	μ	246	÷
7	•	23	↕	39	'	55	7	71	G	87	W	103	g	119	w	135	ç	151	ù	167	º	183	ll	199	ll	215	ll	231	τ	247	≈
8	▣	24	↑	40	(	56	8	72	H	88	X	104	h	120	x	136	ê	152	ÿ	168	¿	184	ll	200	ll	216	ll	232	Φ	248	°
9	○	25	↓	41	)	57	9	73	I	89	Y	105	i	121	y	137	ë	153	Ö	169	¬	185	ll	201	ll	217	ll	233	Θ	249	•
10	◻	26	→	42	*	58	:	74	J	90	Z	106	j	122	z	138	è	154	Ü	170	¬	186	ll	202	ll	218	ll	234	Ω	250	•
11	♂	27	←	43	+	59	;	75	K	91	[	107	k	123	{	139	ï	155	Ç	171	½	187	ll	203	ll	219	ll	235	δ	251	√
12	♀	28	└	44	,	60	<	76	L	92	\	108	l	124		140	î	156	£	172	¼	188	ll	204	ll	220	ll	236	∞	252	n
13	♪	29	↔	45	-	61	=	77	M	93	]	109	m	125	}	141	ì	157	¥	173	;	189	ll	205	=	221	ll	237	φ	253	²
14	🎵	30	▲	46	.	62	>	78	N	94	^	110	n	126	~	142	Ä	158	Ps	174	«	190	ll	206	ll	222	ll	238	€	254	■
15	☼	31	▼	47	/	63	?	79	O	95	_	111	o	127	△	143	Å	159	f	175	»	191	ll	207	ll	223	ll	239	∩	255	ÿ

### 5.0-1

```
#include<iostream>// 字符转编码
using namespace std;
int main() {
    char c;
    int a;
    cin>>c;
    cout<<c<<endl;
    a=c;
    cout<<a<<endl;
    cout<<int(c); //2 种转变方法
    return 0;
}
```

```
r
114
114
```

### 5.0-2

```
#include<iostream>// 编码转字符
using namespace std;
int main() {
    char c;
    int a;
    cin>>a;
    cout<<a<<endl;
    c=a;
    cout<<c<<endl;
    cout<<char(a); //2 种转变方法
    return 0;
}
```

```
114
r
r
```

### #include<iostream>//5.0-3 字符加减

```
#include<cstring>
using namespace std;
int main() {
    char a='a';
    a=a+1;
    cout<<a;
    return 0;
}
```

```
b
```

#### 5.0-4

```
#include<cstdio>//c 语言输入输出字符, ASCII 码
using namespace std;
int main()
{
    char ch1, ch2;
    scanf("%c %c", &ch1, &ch2);
    printf("%c %c\n", ch1, ch2); // 换行 \n
    printf("%d %d", ch1, ch2); //ASCII 码
    return 0;
}
```

```
m n
m n
109 110
```

#### 5.0-5

```
#include<iostream>// 字符存储
using namespace std;
int main()
{
    char c='rr';// 只能保存一个字符
    cout<<c<<endl;
    return 0;
}
```

```
r
```

#### 5.0-6

```
#include<iostream>// 字符存储
using namespace std;
int main()
{
    char c='/0';// 空字符
    char d=' ';// 空格
    cout<<c<<endl; // 空字符输出为零
    cout<<d<<endl; // 空格输出的时候看不见, 但是确实存在
    cout<<int(d); // 空格转换成编码, 就可以看见了
    return 0;
}
```

```
0
32
```

## 5.0-7

`#include<iostream>` // 转义字符, 用于处理一些特殊字符

`using namespace std;`

`int main()`

`{`

`char c='\'';` // 以 \ 开头表示转义字符

`char d='\n';` // 表示换行

`char e='\'';`

`char f='\\';`

`cout<<c;`

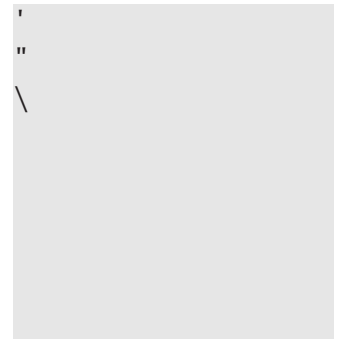
`cout<<d;`

`cout<<e<<endl;`

`cout<<f<<endl;`

`return 0;`

`}`

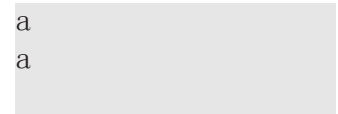


## 字符、字符数组输入输出比较

### 5.1-1

可以输入单个字符

```
#include<iostream>//c++ 输入输出字符
using namespace std;
int main()
{
    char s;
    cin>>s;
    cout<<s;
    return 0;
}
```

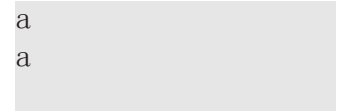


a  
a

### 5.1-2

可以输入单个字符

```
#include<stdio>//c 语言输入输出字符
using namespace std;
int main()
{
    char s;
    scanf("%c",&s);
    printf("%c",s);
    return 0;
}
```



a  
a

### 5.1-3

可以输入多个字符，但不能输入空格

```
#include<iostream>//c++ 输入输出字符数组
using namespace std;
int main()
{
    char s[200];
    cin>>s;
    cout<<s;
    return 0;
}
```

### 5.1-4

`#include<cstdio>`//c 语言输入输出字符数组

```
using namespace std;
int main()
{
    char s[200];
    scanf("%s", s);
    printf("%s", s);
    return 0;
}
```

可以输入多个字符，但不能输入空格


### 附录: getchar(仅做了解, 很少使用)

### 5.1-5

`#include<cstdio>`

```
using namespace std;
int main()
{
    char ch;
    ch=getchar();
    putchar(ch);
    return 0;
}
```

可以输入单个字符



```
ab cdef
a
```

### 5.1-6

`#include<cstdio>`

`#include<cstring>`

```
using namespace std;
int main()
{
    char ch='g';
    putchar(ch);
    putchar('f');
    putchar(86);//ASCII 码
    return 0;
}
```

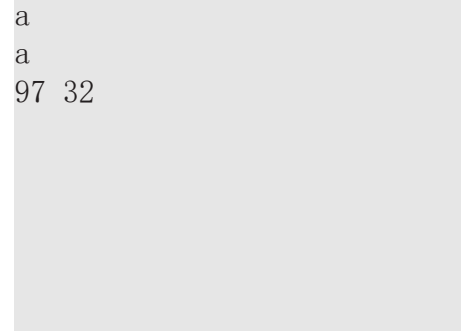


```
gfV
```

## 5.1-7

```
#include<stdio>
using namespace std;
int main()
{
    char ch1, ch2;
    ch1=getchar();
    ch2=getchar();
    putchar(ch1);
    putchar(ch2);
    printf("\n");// 换行 \n
    printf("%d %d", ch1, ch2);
    return 0;
}
```

本题输入 a 和空格，从输出字符只能看见 a 输出，无法观察空格输出，但是转换成 ASCII 码我们可以观察到空格的存在（空格的 ASCII 码是 32）。



```
a
a
97 32
```



## 输入空格

5.2-1// 仅做介绍, gets 已经弃用, 不要用 gets

```
#include<cstdio>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char s[200];
```

```
    gets(s);
```

```
    puts(s);
```

```
    return 0;
```

```
}
```

可以输入多个字符,  
可以输入空格

5.2-2

```
#include<stdio.h>// 仅做参考, 使用比较麻烦
```

```
int main()
```

```
{
```

```
    char s[20]; // 长度为 19, 末尾是 '\0' 的字符数组
```

```
    fgets(s, 5, stdin); // 从输入流 stdin 即输入缓冲区中读取 5 个字符到字符数组 s
```

```
    printf("%s\n", s);
```

```
    return 0;
```

```
}
```

可以输入多个字符, 可以输入空格。由于受到参数 5 和末尾是 \0 的限制, 只能输入输出 ab 空格 c, 一共 4 个字符。

```
ab cdef  
ab c
```

5.2-3

```
#include<iostream>// 常用 getline
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char s[200];
```

```
    cin.getline(s, 5);
```

```
    cout<<s;
```

```
    return 0;
```

```
}
```

可以输入多个字符, 可以输入空格。由于受到参数 5 和末尾是 \0 的限制, 只能输入输出 ab 空格 c, 一共 4 个字符。

```
ab cdef  
ab c
```

## 5.2-4

```
#include<iostream>// 用 scanf 输入空格
#include<cstdio>
using namespace std;
int main()
{
    char s[100];
    char ch;
    int i=0;
    while (scanf('%c', &ch) && ch != '\n')
    {
        s[i++] = ch;
    }
    s[i] = '\0'; // 删除这行，会出错，观察第二张图片。
    cout << s << endl;
    return 0;
}
```

## 5.2-5

```
#include<iostream>// 用 getchar 输入空格
#include<stdio.h>
using namespace std;
int main()
{
    char s[100];
    char ch;
    int i=0;
    while ((ch=getchar()) != '\n')
    {
        s[i++] = ch;
    }
    s[i] = '\0'; // 字符串末尾字符赋值成 '\0'
    cout << s << endl;
    return 0;
}
```

scanf 输入遇到 \n 会停止输入，但是输出时候会以 \0 判断是不是字符数组的末尾。所以，把最后字符改成 \0，才不会出错。

```
356 789
356 789
```

```
123 456 789
123 456 789k
```

getchar 输入遇到 \n 会停止输入，但是输出时候会以 \0 判断是不是字符数组的末尾。所以，把最后字符改成 \0，才不会出错。

```
356 789
356 789
```

## 字符数组基础知识

5.3-1

```
#include<iostream>// 字符数组没有赋值
using namespace std;
int main()
{
    char a[10];// 没有赋值, 显示是随机数据
    int i;
    for (i=0;i<10;i++)
        cout<<a[i]<<" ";

    cout<<endl;

    char b[20]={0};// 没有赋值
    char c[20]={};// 没有赋值
    cout<<b<<endl;// 不显示
    cout<<c<<endl;// 不显示

    return 0;
}
```

### 5.3-2

```
#include<iostream> // 字符数组赋值的几种办法
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char a[10];
```

```
    a[0]='w' ;
```

```
    a[1]='e' ;
```

```
    a[2]='\0' ; // 字符数组以 \0 作为结尾
```

```
    cout<<a<<endl ; // 字符数组可以这样输出，整数数组不可以
```

```
    char b[10] ={'w','o','r','l','d'};
```

```
    // 如果最后没写 \0，程序自动补充 \0
```

```
    cout<<b<<endl ;
```

```
    char c[] ={'w','o','r','l','d','\0'};
```

```
    // 没设置长度，最后字符写 \0，如果不写，容易出错。
```

```
    cout<<c<<endl ;
```

```
    char d[] ="world"; // 没写 \0，程序自动补充 \0
```

```
    cout<<d<<endl ;
```

```
    char e[] ={'w','o','r','l','d','\0','w','w','w'};
```

```
    //          \0 后面的不生效，不显示
```

```
    cout<<e<<endl ;
```

```
    return 0;
```

```
}
```

```
we  
world  
world  
world  
world
```

### 5.3-3

```
#include<iostream>// 二维字符数组赋值
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int i, j;
```

```
    char ch[3][4]={"abc", "def", "567"};
```

```
    for (i=0; i<3; i++)
```

```
    {
```

```
        for (j=0; j<4; j++)// 为什么是4? 但却只显示3个字母? 因为末尾是 \0
```

```
        {
```

```
            cout<<ch[i][j]<<" ";
```

```
        }
```

```
        cout<<endl;
```

```
    }
```

```
    return 0;
```

```
}
```

```
a b c  
d e f  
5 6 7
```

### 5.3-4

```
#include<iostream>// 字符数组长度
#include<cstring>// 没有这句, strlen 不运行
using namespace std;
int main()
{
    char a[10];
    a[0]='w' ;
    a[1]='e' ;
    a[2]='\0' ;// 字符数组以 \0 作为结尾
    int i;

    cout<<sizeof(a)<<endl;// 计算变量占用字节数
    cout<<strlen(a) ; // 字符数组实际长度
    cout<<endl ;

    char d[] ="world";// 没写 \0, 程序自动补充 \0
    cout<<d<<" " <<sizeof(d)<<" " <<strlen(d) ;
    //sizeof 测量的长度是 6, 因为自动加了 \0;
    //strlen 测量的长度是 5, 实际长度, 不计算 \0;
    return 0;
}
```

```
10
2
world 6 5
```

### 5.3-5

`#include<iostream>`// 遍历字符数组，字符数组默认以 `\0` 作为结束

`#include<cstring>`

`using namespace std;`

`int main()`

`{`

`char a[100]="wo\0rld";`

`char b[100]={'w','o','\0','r','l','d'};`

`cout<<a<<endl;`// 字符数组以 `\0` 结束

`cout<<b<<endl;`// 字符数组以 `\0` 结束

`cout<<strlen(b)<<endl;`// 长度测量到 `\0` 结束

`int i;`

`for (i=0;i<strlen(b);i++)`

`cout<<b[i]<<" ";`// 字符数组以 `\0` 结束

`cout<<endl;`

`for (i=0;b[i]!='\0';i++)`

`cout<<b[i]<<" ";`// 字符数组以 `\0` 结束

`cout<<endl;`

`for (i=0;i<=99;i++)`

`cout<<b[i]<<" ";`// 显示完整的字符数组

`cout<<endl;`

`return 0;`

`}`

```
wo
wo
2
w o
w o
w o   r l d
```

### 5.3-6

```
#include<iostream> // 字符数组输入输出空格
#include<cstring>
using namespace std;
int main()
{
    char a[20];
    fgets(a, 19, stdin); // gets 已经弃用, 不可使用, 容易出错, 用 fgets。
    int i;
    cout<<a<<endl;

    for (i=0; i<strlen(a); i++)        cout<<a[i]<<" "; // 字符数组以 \0 结束
    cout<<endl;

    for (i=0; a[i]!='\0'; i++)        cout<<a[i]<<" "; // 字符数组以 \0 结束
    cout<<endl;

    for (i=0; i<=19; i++)
        cout<<a[i]<<" "; // 显示完整的字符数组, 后面没有的字符随机显示
    cout<<endl;
    return 0;
}
```

```
rye fgh 5
rye fgh 5

r y e   f g h   5
r y e   f g h   5
r y e   f g h   5
```

### 5.3-7

```
#include<iostream> // 截断字符数组
#include<cstring>
using namespace std;
int main()
{
    char s[10]="abcd fklj";
    cout<<s<<endl; // 第一次输出
    s[4]='\0';
    cout<<s; // 第二次输出
    return 0;
}
```

```
abcd fklj
abce
```



### 5.3-8

```
#include<iostream>// 大小写字母转换
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char c;
```

```
    int a;
```

```
    cin>>c;
```

```
    a=c;
```

```
    if(a>=97&&a<=122)
```

```
        a=a-32;// 如果忘记 32 这个数字, 也可以写 c=c-('a'-'A')
```

```
    else if(a>=65&&a<=90)
```

```
        a=a+32;
```

```
    c=a;
```

```
    cout<<c;
```

```
    return 0;
```

```
}
```

```
r  
R
```

```
R  
r
```

### 5.3-9

```
#include<iostream>// 小写字母转大写
```

```
#include&ltcstring>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char s[10]="abcdef";
```

```
    for(int i=0;s[i]!='\0';i++)
```

```
    {
```

```
        s[i]=s[i]-('a'-'A');
```

```
    }
```

```
    cout<<s;
```

```
    return 0;
```

```
}
```

```
ABCDEF
```

### 5.3-10

```
#include<iostream>// 替换字符
#include<cstring>
using namespace std;
int main()
{
    int n;
    char s[10],s1;
    cin>>s;// 输入字符数组
    cin>>n>>s1;// 把第 n 个字符替换成 s1

    s[n-1]=s1;
    cout<<s;
    return 0;
}
```

```
abcdef
3 8
ab8def
```

### 5.3-11

```
#include<iostream>// 统计小写字母
#include<cstring>
using namespace std;
int main()
{
    char s[200];
    int i,count=0;
    cin>>s;
    for (i=0;s[i];i++)// 结束语句是不常见的写法
    {
        if(s[i]>='a'&&s[i]<='z')// 这个范围都是小写字母
            count++;
    }
    cout<<count;
    return 0;
}
```

```
GkGgN
2
```

### 5.3-12

```
#include<iostream>// 字符数组数字求和
```

```
#include<cstring>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char a[2000]; // 假设输入的数字位数高达几百位，如果作为数字输入，  
                // 数值太大，计算机没法计算。只能用字符数组输入。
```

```
    cin>>a;      // 本题没有空格，所以可以用 cin
```

```
    int i;
```

```
    int s=0;
```

```
    for (i=0;i<strlen(a);i++)
```

```
    {
```

```
        s=s+(a[i]-48); // 数字减去编码，等于数字本身
```

```
    }
```

```
    cout<<s<<endl;
```

```
    s=0;
```

```
    for (i=0;i<strlen(a);i++)
```

```
    {
```

```
        s=s+(a[i]-'0');
```

```
    }
```

```
    cout<<s;
```

```
    return 0;
```

```
}
```

```
12345
```

```
15
```

```
15
```

```

#include<iostream>//5.3-13 查找特定字符
#include<cstring>
using namespace std;
int main() {
    char a[200];
    int i;
    char q;//需要查找的字符
    int p;//特定字符下标
    cin>>a; // 本题没有空格，所以可以用 cin
    cin>>q;
    for (i=0;i<strlen(a);i++)
    {
        if(a[i]==q)
        {
            p=i;
            cout<<p;
        }
    }
    return 0;
}

```

```

world
r
2

```

### 5.3-14

```

#include<iostream>// 字符 数组，整数数组输出比较
#include<cstring>
using namespace std;
int main() {
    char a[200]="world";
    int b[100]={100,200,300};
    int i;
    cout<<a<<endl;
    cout<<b<<endl;// 下标为 0 的元素的地址
    cout<<&b<<endl;//&, 取地址运算符。获取变量 b 的地址
    cout<<&b[0]<<endl;//&, 取地址运算符。获取变量 b 下标 0 的元素的地址
    for (i=0;i<3;i++)        cout<<b[i]<<" ";
    return 0;
}

```

```

world
0x6ffc70
0x6ffc70
0x6ffc70
100 200 300

```

## 字符数组常用函数

5.4-1-a

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char s[10];
    cin>>s;
    int len=strlen(s); // 常用函数，获取字符数组长度
    cout<<len;//len 或者 l 常用来表示字符数组长度
    return 0;
}
```

abc

3

5.4-1-b

```
#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char s[10];
    cin>>s;

    cout<<strlen(s);
    return 0;
}
```

abc

3

#### 5.4-2

```
#include<iostream>// 比较函数
#include<cstring>
using namespace std;
int main()
{
    char s1[100], s2[100];
    cin>>s1>>s2;
    int ret=strcmp(s1, s2);// 比较函数
    cout<<ret;// 如果s1>s2, 返回值1; 如果s1<s2, 返回值-1; 如果相等, 返回值等于0。
    return 0;
}
```

```
abc def
-1
```

#### 5.4-3

```
#include<iostream>//n 个字符比较函数
#include<cstring>
using namespace std;
int main()
{
    int n;
    char s1[100], s2[100];
    cin>>s1>>s2>>n;
    int ret=strncmp(s1, s2, n);// 比较前 n 个字符
    cout<<ret;
    return 0;
}
```

```
abcerg abc567
3
0
```

5. 4-4

```
#include<iostream>// 复制函数
#include<cstring>
using namespace std;
int main()
{
    char s1[100], s2[100];
    cin>>s1>>s2;
    strcpy(s1, s2);
    cout<<s1;
    return 0;
}
```

```
abcdef tyu
tyu
```

```
abc def
def
```

5. 4-5

```
#include<iostream>//n 个字符复制函数
#include<cstring>
using namespace std;
int main()
{
    char s1[100], s2[100];
    cin>>s1>>s2;
    strncpy(s1+3, s2, 2);//s2 前两个字符复制到 s1 第 3 个字符之后
    cout<<s1;
    return 0;
}
```

```
abcdef 123456
abc12f
```

5.4-6

```
#include<iostream>// 链接函数
#include<cstring>
using namespace std;
int main()
{
    char s1[100], s2[100];
    cin>>s1>>s2;
    strcat(s1, s2);
    cout<<s1;
    return 0;
}
```

```
abc def
abcdef
```

5.4-7

```
#include<iostream>//n 个字符连接函数
#include<cstring>
using namespace std;
int main()
{
    char s1[100], s2[100];
    cin>>s1>>s2;
    strncat(s1, s2, 2); //s2 的前 2 个字符连接到 s1 后
    cout<<s1;
    return 0;
}
```

```
abc 12345
abc12
```



## string 基础知识

### 5.5-1

```
#include<iostream>// 字符串输入输出（不能录入空格）
using namespace std;
int main()
{
    string s;
    cin>>s;
    cout<<s;
    return 0;
}
```

```
abc def
abc
```

### 5.5-2

```
#include<iostream>// 字符串输入空格
using namespace std;
int main()
{
    string s;
    getline(cin, s);
    cout<<s;
    return 0;
}
```

```
abc def
abc def
```

### 5.5-3

```
#include<iostream>// 输出单个字符、输出整个数组
using namespace std;
int main()
{
    string s="abc def";
    cout<<s[2]<<endl;
    cout<<s;
    return 0;
}
```

```
c
abc edf
```

#### 5.5-4

```
#include<iostream>// 正向输出字符
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    string s="abc def";
```

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

```
    {
```

```
        cout<<s[i]<<" ";
```

```
    }
```

```
    return 0;
```

```
}
```

```
a b c   d e f
```

#### 5.5-5

```
#include<iostream>// 反向输出字符
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    string s="abc def";
```

```
    for (int i=s.size();i>=0;i--)
```

```
    {
```

```
        cout<<s[i]<<" ";
```

```
    }
```

```
    return 0;
```

```
}
```

```
f d e   c b a
```

### 5.5-6

```
#include<iostream>// 字符串赋值
```

```
using namespace std;
```

```
int main()
```

```
{  
    string s1, s2;  
    s1="abc";  
    s2=s1;  
    cout<<s2<<endl;  
    return 0;  
}
```

abc

### 5.5-7

```
#include<iostream>// 字符串合并
```

```
using namespace std;
```

```
int main()
```

```
{  
    string s1, s2, s3;  
    s1="abc";  
    s2="def";  
    s3=s1+s2;  
    cout<<s3;  
    return 0;  
}
```

abcdef

### 5.5-8

```
#include<iostream>// 正确赋值
```

```
#include<cstring>
```

```
using namespace std;
```

```
int main()
```

```
{  
    string a="world";  
    a[0]='h';  
    a[1]='h';  
    cout<<a;  
    return 0;  
}
```

hhrld

### 5.5-9

```
#include<iostream>// 错误赋值
#include<cstring>
using namespace std;
int main()
{
    string a; // 现在 a 是空的
    a[0] = 'w'; // 字符串没有设置长度，字符串是空的，
    a[1] = 'e'; // 不存在下标，所以无法用这种方式添加字符
    cout<<a; // 没有输出，表明这个赋值的办法是错误的
    return 0;
}
```

### 5.5-10

```
#include<iostream>// 字典码比较
#include<cstring>
using namespace std;
int main() // 从第一个字符开始比较，若有结果，比较结束
{
    string a="world";
    string b="hello";
    if(a>b)
        cout<<a<<endl;
    else
        cout<<b<<endl;

    string c="123";
    string d="56";
    if(c>d)
        cout<<c<<endl;
    else
        cout<<d<<endl;

    return 0;
}
```



```
world
56
```

### 5.5-11

```
#include<iostream>// 字符替换
#include<string>
using namespace std;
int main()
{
    string s;
    cin>>s;
    int len=s.length();
    for (int i=0;i<len;i++)
    {
        if(s[i]=='a')
            s[i]='A';
        if(s[i]=='b')
            s[i]='B';
        if(s[i]=='c')
            s[i]='C';
    }
    cout<<s;
    return 0;
}
```

abc  
ABC

### 5.5-12

```
#include<iostream>// 输出字母表
#include<cstring>
using namespace std;
int main()
{
    string a;
    char b;
    for (b='a';b<='z';b++)
        a=a+b;
    cout<<a;
    return 0;
}
```

abcdefghijkl  
mnopqrstuvwxyz

## string 常用函数

5.6-1

```
#include<iostream>// 求字符串长度
using namespace std;
int main()
{
    string s="abcdef";
    int len=s.size();// 求字符串长度函数
    int l=s.length();// 求字符串长度函数
    cout<<len<<endl;
    cout<<l;
    return 0;
}
```

```
6
6
```

5.6-2

```
#include<iostream>// 截取函数 substr
using namespace std;
int main()
{
    string a="hello world! hello world! ";
    string b=a.substr(6);// 从下标 6 开始截取, 到最后
    cout<<b<<endl;

    string c=a.substr(6,7);// 从下标 6 开始截取, 截取 7 个字符
    cout<<c;
    return 0;
}
```

```
world! hello world!
world!
```

### 5.6-3

```
#include<iostream>// 查找函数 find
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    string a="hello world! hello world! ";
```

```
    cout<<a.find("wo")<<endl;    // 查找 wo 出现的第一次下标
```

```
    cout<<a.find("wo", 11)<<endl; // 从下标 11 之后开始查找 wo 的下标
```

```
    int p=a.find("wo");
```

```
    cout<<p<<endl;    //p 是下标
```

```
    string b="www";
```

```
    int c=a.find(b); // 查找的字符，找不到，怎么输出？
```

```
    cout<<c;    // 找不到，输出 -1
```

```
    return 0;
```

```
}
```

```
6  
20  
6  
-1
```

5.6-4

```
#include<iostream>// 连续查找
using namespace std;
int main()
{
    string a,b;
    getline(cin,a);
    getline(cin,b);
    int p=a.find(b); //p 是下标
    if(p!=-1)
    {
        cout<<-1;
    }
    else
    {
        while(p!=-1)
        {
            cout<<p+1<<endl;
            p=a.find(b,p+1);// 继续查找
        }
    }
    return 0;
}
```

```
hello world! hello world!
wo
7
21
```



### 5.6-5

```
#include<iostream>// 连续查找
using namespace std;
int main()
{
    string a, b, x;
    getline(cin, a);
    getline(cin, b);
    int i;
    for (i=0; i<a.size(); i++)
    {
        x=a.substr(i, b.size());
        //cout<<x<<" ";
        if(x==b)
            cout<<i<<" ";
    }
    return 0;
}
```

```
hello world!  hello world!
wo
6
20
```

### 5.6-6

```
#include<iostream>// 删除子字符串 erase
using namespace std;
int main()
{
    string a="hello world!  hello world! ";
    a.erase(0, 2);// 从下标 0 开始删除, 删除长度 2
    cout<<a<<endl;
    string b="hello world!  hello world! ";
    b.erase(5);// 从下标 5 开始删除, 全部删除
    cout<<b<<endl;
    return 0;
}
```

```
llo world!  hello world!
hello
```

### 5.6-7

```
#include<iostream>// 插入函数 insert
using namespace std;
int main() {
    string a="hello world! hello world! ";
    a.insert(5, "***");// 从下标 5 插入
    cout<<a<<endl;
    return 0;
}
```

```
hello*** world! hello
world!
```

### 5.6-8

```
#include<iostream>// 替换函数 replace
using namespace std;
int main() {
    string a="hello world! hello world! ";
    a.replace(6, 5, "***");// 从下标 5 开始替换, 替换 5 个字符
    cout<<a<<endl;
    return 0;
}
```

```
hello ***! hello world!
```

### 5.6-9

```
#include<iostream>// 连续替换
using namespace std;
int main()
{
    string a="hello world! hello world! ";
    string b1, b2;
    cin>>b1>>b2;
    int p=a.find(b1);
    while(p!=-1)
    {
        a.replace(p, b1.size(), b2);
        p=a.find(b1);
    }
    cout<<a;
    return 0;
}
```

```
wo
**
hello **rld! hello **rld!
```

## 5.6-10

```
#include<iostream>// 字符类型函数
using namespace std;
int main()
{
    char a, b, c, d;
    cin>>a;

    if(isalpha(a))// 判断是否为字母
        cout<<" 字母 "<<endl;
    if(islower(a))// 判断是否小写字母
        cout<<" 小写字母 "<<endl;
    if(isupper(a))// 判断是否大写字母
        cout<<" 大写字母 "<<endl;
    if(isdigit(a))// 判断是否数字
        cout<<" 数字 "<<endl;

    return 0;
}
```

a  
字母  
小写字母

B  
字母  
大写字母

56  
数字

## 5.6-11

```
#include<iostream>// 大小写转换
using namespace std;
int main()
{
    char a='a', b='B';

    cout<<toupper(a)<<endl;
    cout<<tolower(b)<<endl;

    a=toupper(a);// 小写转大写
    b=tolower(b);// 大写转小写
    cout<<a<<endl;
    cout<<b<<endl;
    return 0;
}
```

65  
98  
A  
b

## 5.6-12

`#include<iostream>`// 头尾指针, 请先学习 stl 中的 `sort` 函数和 `reverse` 函数

`#include<algorithm>`

`using namespace std;`

`int main()`

`{`

`string a="cabde";`

`sort(a.begin(), a.end());`// 从头开始, 从小排列

`int i;`

`cout<<a<<endl;`

`for (i=0; i<5; i++)`

`cout<<a[i]<<" ";`

`cout<<endl;`

`string b="*cabde*";`

`sort(b.begin()+1, b.end()-1);`// 从下标 1 开始, n-2 个数据参与排序

`cout<<b<<endl;`

`return 0;`

`}`

```
abcde
a b c d e
*abcde*
```

## 5.6-13

`#include<iostream>`// 头尾指针, 请先学习 stl 中的 `sort` 函数和 `reverse` 函数

`#include<algorithm>`

`using namespace std;`

`int main()`

`{`

`string a="abcde";`

`reverse(a.begin(), a.end());`// 从头开始, 5 个数据反转

`int i;`

`cout<<a<<endl;`

`for (i=0; i<5; i++)`

`cout<<a[i]<<" ";`

`cout<<endl;`

`string b="*abcde$";`

`reverse(b.begin()+1, b.end()-1);`// 从下标 1 开始, n-2 个数据反转

`cout<<b;`

`return 0;`

`}`

```
edcba
e d c b a
*edcba$
```

## 5.6-14

```
#include<iostream>// 数值、字符串互转
using namespace std;
int main()// 设置: 工具 - 编译选项, -std=c++11
{
    string a="56";
    int x1=stoi(a); // 字符串转换 int
    cout<<x1<<endl;

    string b="569999999999999";
    long long x2=stoll(b); // 字符串转换 long long
    cout<<x2<<endl;

    string c="16.77";
    float x3=stof(c); // 字符串转换 float
    cout<<x3<<endl;

    int d=9999;
    string x4=to_string(d); //int 转换字符串
    cout<<x4<<endl;

    double e=9999.999;
    string x5=to_string(e); //double 转换字符串
    cout<<x5<<endl;

    return 0;
}
```

```
56
569999999999999
16.77
9999
9999.999000
```