

第一章 顺序结构

第9课 字符类型 char

《信息学奥赛一本通·编程启蒙 C++版》

如果你在 C++ 程序中看到如下表中的这三个 1，它们是有很大区别的。

表 1-2 三个 1

内容	特点	表示
1	什么都没有加	数字 1
'1'	加了单引号	字符 '1'
"1"	加了双引号	字符串 "1"

如果在 C++ 程序中看到 a、'a' 以及 "a"，你知道它们三者的区别吗？是的，a 表示一个变量；'a' 表示字符；"a" 表示字符串，只不过这个字符串里面只有一个字符。

字符常量是由单个字符组成，所有字符采用 ASCII (American Standard Code for Information Interchange 美国信息交换标准代码) 编码，ASCII 编码共有 128 个字符(表 1-3)。在程序中，通常用一对单引号将单个字符括起来表示一个字符常量。如：'a'，'A'，'0' 等。如字符 A 的序号是 65，字符 a 的序号是 97，字符 '0' 的序号的 48。

ASCII control characters

00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

ASCII printable characters

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

定义字符变量的语法格式为：`char` 变量表

`char` 是 `character` 的缩写。我们可以将 ASCII 码值或者字符常量赋值给字符变量。我们还可以用 `cin` 将字符通过键盘输入到字符变量中。字符还可以参加算术、比较大小等运算，在参加运算的时候，字符等价于其对应的 ASCII 码。

【例 9.1】输出字符三角形

【题目描述】

现在给定一个字符，希望你用它构造一个底边长 5 个字符，高 3 个字符的等腰字符三角形。

【输入格式】

输入为一个字符。

【输出格式】

该字符构成的等腰三角形，底边长 5 个字符，3 个字符。

【样例输入】

*

【样例输出】

*

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. char c;
4. int main(){
5.     cin>>c;
6.     cout<<"  "<<c<<endl;
7.     cout<<"  "<<c<<c<<c<<endl;
8.     cout<<c<<c<<c<<c<<c<<endl;
9.     return 0;
10. }
```

【例 9.2】打印字符

【题目描述】

我们知道每个字符都有一个对应的 ASCII 码。现在输入一个 ASCII 码，要求你输出对应的字符。。

【输入格式】

一个整数，即字符的 ASCII 码，保证对应的字符为可见字符。

【输出格式】

一行，包含相应的字符。

【样例输入】

65

【样例输出】

A

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. int n;
4. char c;
5. int main(){
6.     cin>>n;
7.     c=n;
8.     cout<<c;
9.     return 0;
10.}
```

【例 9.3】小写字母转大写字母(趣味编程)

【题目描述】

输入一个小写字母，则变为大写输出。

【输入格式】

输入为一个小写字符。

【输出格式】

对应的大写字母。

【样例输入】

a

【样例输出】

A

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. char c;
4. int main(){
5.     cin>>c;
6.     c=c-32;
7.     cout<<c;
8.     return 0;
9. }
```

练 9.1 字符菱形

【题目描述】

给定一个字符，用它构造一个对角线长 5 个字符，倾斜放置的菱形。

【输入格式】

输入只有一行， 包含一个字符

【输出格式】

该字符构成的菱形

【样例输入】

*

【样例输出】

*

*

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. char c;
4. int main(){
5.     cin>>c;
6.     cout<<" "<<c<<endl;
7.     cout<<" "<<c<<c<<c<<endl;
8.     cout<<c<<c<<c<<c<<c<<endl;
9.     cout<<" "<<c<<c<<c<<c<<endl;
10.    cout<<" "<<c<<endl;
11.    return 0;
12.}
```

练 9.2 打印 ASCII 码

【题目描述】

输入一个除空格以外的可见字符，输出其 ASCII 码。

【输入格式】

一个除空格以外的可见字符

【输出格式】

一个十进制整数，即该字符的 ASCII 码

【样例输入】

A

【样例输出】

65

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. char c;
4. int n;
5. int main(){
6.     cin>>c;
7.     n=c;
8.     cout<<n;
9.     return 0;
10.}
```

练 9.3 加四密码

【题目描述】

要将 China 译成密码，译码规律是：用原来字母后面的第 4 个字母代替原来的字母。例如，字母 A 后面第 4 个字母是 E。E 代替 A。因此，China 应译为 G1mre。请编一程序，用赋初值的方法使 c1、c2、c3、c4、c5 五个变量的值分别为，C、h、i、n、a，经过运算，使

c1、c2、c3、c4、c5 分别变为 G、l、m、r、e，并输出。

【输入格式】

输入长度为 5 的字符串（保证字母后面的第 4 个字符存在）

【输出格式】

输出加密后的结果

【样例输入】

China

【样例输出】

G1mre

【代码实现】

```
1. #include<bits/stdc++.h>
2. using namespace std;
3. char a,b,c,d,e;
4. int main(){
5.     cin>>a>>b>>c>>d>>e;
6.     a=a+4;
7.     b=b+4;
8.     c=c+4;
9.     d=d+4;
10.    e=e+4;
11.    cout<<a<<b<<c<<d<<e;
12.    return 0;
13.}
```

谢谢!

—