



西北工业大学
NORTHWESTERN POLYTECHNICAL UNIVERSITY

C程序设计 Programming in C



1011014

主讲：姜学锋，计算机学院

编写程序语句

4、单个字符输入输出

3.2 输入与输出

- ▶ 所谓输入是指从外部输入设备（如键盘、鼠标等）向计算机输入数据，输出是指从计算机向外部输出设备（如显示器、打印机等）输出数据。

3.2 输入与输出

输入输出工作原理



3.2 输入与输出

- ▶ C语言**输入输出操作是通过函数实现的**。不同的函数能够处理形式多样的输入输出操作，支持不同的输入输出设备。
- ▶ C语言标准库中定义了标准输入输出函数，以标准的终端设备（键盘和显示器）为输入输出设备，有字符输出putchar、字符输入getchar、格式输出printf、格式输入scanf等函数。

3.2 输入与输出

- ▶ 若在程序中调用标准输入输出函数，应该用文件包含命令将头文件stdio.h包含到程序中，命令形式为：

```
#include <stdio.h>
```

- ▶ 因此，C语言输入输出操作本质上是函数调用语句。

3.2.1 字符输入与输出

- ▶ 1. 字符输出putchar函数
- ▶ putchar函数的作用是向显示终端输出一个字符，一般形式为：

```
putchar(c);
```

- ▶ 其中参数c为整型，使用低八位的值，输出的字符是c值对应的ASCII符号。函数调用时，c可以是常量、变量或表达式，可以是整型数据、字符型数据或转义字符。

3.2.1 字符输入与输出

- ▶ putchar函数可以直接输出ASCII码对照表中可显示的字符（ASCII值为0x20~0x7f），控制字符（ASCII值为0x00~0x1f）的输出有特殊的含义。
- ▶ 例如'\n'输出换行符，使光标移到下一行的开头，'\r'输出回车符，使光标回到本行开头。

附录A ASCII码对照表																								
DEC	HEX	字符	控制字符	DEC	HEX	字符	DEC	HEX	字符	DEC	HEX	字符	DEC	HEX	字符	DEC	HEX	字符	DEC	HEX	字符	DEC	HEX	字符
0	00	(null)	NUL	32	20	空格	64	40	@	96	60	`	128	80	Ç	160	A0	à	192	C0	À	224	E0	α
1	01	☺	SOH	33	21	!	65	41	A	97	61	a	129	81	Û	161	A1	á	193	C1	Á	225	E1	β
2	02	☻	STX	34	22	"	66	42	B	98	62	b	130	82	é	162	A2	â	194	C2	Â	226	E2	Γ
3	03	♥	ETX	35	23	#	67	43	C	99	63	c	131	83	â	163	A3	û	195	C3	Û	227	E3	π
4	04	♦	EOT	36	24	\$	68	44	D	100	64	d	132	84	ä	164	A4	ü	196	C4	Ü	228	E4	Σ
5	05	♣	ENQ	37	25	%	69	45	E	101	65	e	133	85	å	165	A5	ñ	197	C5	Ñ	229	E5	σ
6	06	♠	ACK	38	26	&	70	46	F	102	66	f	134	86	ä	166	A6	ª	198	C6	ª	230	E6	μ
7	07	•	BEL	39	27	'	71	47	G	103	67	g	135	87	ç	167	A7	º	199	C7	º	231	E7	τ
8	08	▣	BS	40	28	(72	48	H	104	68	h	136	88	è	168	A8	¿	200	C8	¿	232	E8	Ø
9	09	○	TAB	41	29)	73	49	I	105	69	i	137	89	ë	169	A9	¬	201	C9	¬	233	E9	Θ
10	0A	☼	LF	42	2A	*	74	4A	J	106	6A	j	138	8A	è	170	AA	¬	202	CA	¬	234	EA	Ω
11	0B	☾	VT	43	2B	+	75	4B	K	107	6B	k	139	8B	ï	171	AB	½	203	CB	½	235	EB	δ
12	0C	♀	FF	44	2C	,	76	4C	L	108	6C	l	140	8C	î	172	AC	¾	204	CC	¾	236	EC	∞
13	0D	♂	CR	45	2D	-	77	4D	M	109	6D	m	141	8D	í	173	AD	¼	205	CD	¼	237	ED	ø
14	0E	♂	SO	46	2E	.	78	4E	N	110	6E	n	142	8E	ä	174	AE	«	206	CE	«	238	EE	€
15	0F	♀	SI	47	2F	/	79	4F	O	111	6F	o	143	8F	Å	175	AF	»	207	CF	»	239	EF	∫
16	10	▶	DLE	48	30	0	80	50	P	112	70	p	144	90	É	176	BO	≡	208	DO	≡	240	FO	≡
17	11	◀	DC1	49	31	1	81	51	Q	113	71	q	145	91	æ	177	B1	≡	209	D1	≡	241	F1	±
18	12	↑	DC2	50	32	2	82	52	R	114	72	r	146	92	Æ	178	B2	≡	210	D2	≡	242	F2	≥
19	13		DC3	51	33	3	83	53	S	115	73	s	147	93	Ø	179	B3	≡	211	D3	≡	243	F3	≤
20	14	¶	DC4	52	34	4	84	54	T	116	74	t	148	94	Ö	180	B4	≡	212	D4	≡	244	F4	∫
21	15	§	NAK	53	35	5	85	55	U	117	75	u	149	95	ö	181	B5	≡	213	D5	≡	245	F5	∫
22	16	—	SYN	54	36	6	86	56	V	118	76	v	150	96	Û	182	B6	≡	214	D6	≡	246	F6	÷
23	17	↓	ETB	55	37	7	87	57	W	119	77	w	151	97	ü	183	B7	≡	215	D7	≡	247	F7	≈
24	18	↑	CAN	56	38	8	88	58	X	120	78	x	152	98	ý	184	B8	≡	216	D8	≡	248	F8	•
25	19	↓	EM	57	39	9	89	59	Y	121	79	y	153	99	ÿ	185	B9	≡	217	D9	≡	249	F9	•
26	1A	→	SUB	58	3A	:	90	5A	Z	122	7A	z	154	9A	Ü	186	BA	≡	218	DA	≡	250	FA	•
27	1B	←	ESC	59	3B	;	91	5B	[123	7B	{	155	9B	ƒ	187	BB	≡	219	DB	≡	251	FB	√
28	1C	↵	FS	60	3C	<	92	5C	\	124	7C		156	9C	£	188	BC	≡	220	DC	≡	252	FC	²
29	1D	↵	GS	61	3D	=	93	5D]	125	7D	}	157	9D	¥	189	BD	≡	221	DD	≡	253	FD	³
30	1E	↵	RS	62	3E	>	94	5E	^	126	7E	~	158	9E	℔	190	BE	≡	222	DE	≡	254	FE	■
31	1F	↵	US	63	3F	?	95	5F	_	127	7F	◊	159	9F	ƒ	191	BF	≡	223	DF	≡	255	FF	■

DEC: 十进制ASCII值; HEX: 十六进制ASCII值; 128~255为扩展ASCII码。

DEC: 十进制ASCII值; HEX: 十六进制ASCII值; 128~255为扩展ASCII码。

3.2.1 字符输入与输出



【例3.1】

使用putchar输出字符。

3.2.1 字符输入与输出

例3.1

```
1 #include <stdio.h>
2 int main()
3 {
4     char a='C', b=6;
5     putchar(a); //字符型变量, 输出 C
6     putchar(b+'0'); //整型表达式, 输出 6
7     putchar('\110'); //转义字符(八进制), 输出 H
8     putchar('\n'); //转义字符(换行), 输出 换行
9     return 0;
10 }
```

3.2.1 字符输入与输出

例3.1

程序运行屏幕



3.2.1 字符输入与输出

- ▶ 2. 字符输入getchar函数
- ▶ getchar函数的作用是从键盘终端输入一个字符，一般形式为：

```
getchar();
```

3.2.1 字符输入与输出

- ▶ getchar函数没有参数，函数返回值为输入的字符。通常将getchar的返回值赋给一个字符型变量或整型变量，
- ▶ 示例

```
c=getchar(); //输入字符保存到c中，以便后续能使用它（用c）
```

- ▶ 或者作为表达式的一部分直接使用，例如：

```
putchar( getchar() ); //将输入字符直接输出  
putchar( c=getchar() ); //将输入字符保存到c中，并且输出
```

3.2.1 字符输入与输出

- ▶ getchar函数的输入操作步骤如下：
- ▶ ①检查键盘缓冲区是否有字符；
- ▶ ②若有字符则直接从缓冲区中提取一个字符返回，且缓冲区移向下一个字符；
- ▶ ③若没有字符则getchar等待键盘输入，直到输入回车结束等待，重复步骤①。

3.2.1 字符输入与输出

► 例如执行：

```
1 c=getchar();
```

- getchar将等待键盘输入，如果输入1↵（本书用↵表示键入回车），则键盘缓冲区有两个字符，c提取了字符"1"，键盘缓冲区还留有一个字符"↵"。

3.2.1 字符输入与输出

▶ 又如执行：

```
1 c1=getchar();  
2 c2=getchar();
```

- 执行第1行时getchar等待键盘输入，如果输入1↵，那么c1提取了字符"1"；执行第2行时由于键盘缓冲区还有一个字符，故第2行不用等待键盘输入，直接提取字符"↵"。

3.2.1 字符输入与输出

- ▶ getchar函数执行时从键盘上可以连续输入多个字符，直到回车为止。输入的多个字符放到键盘缓冲区，一次getchar函数调用会从缓冲区中提取一个字符，直到缓冲区没有字符时才从键盘上输入。

3.2.1 字符输入与输出



如何让正在运行中的程序暂停一下？

有时，程序员出于调试目的希望让程序执行到某一行时停下来，便于观察运行情况，按回车继续执行。

那么在这一行写下面的代码：

```
getchar(); // 由于只是希望停下来，因而不在于返回什么，所以不用接收返回值
```

3.2.1 字符输入与输出



【例3.2】

使用getchar输入字符。

3.2.1 字符输入与输出

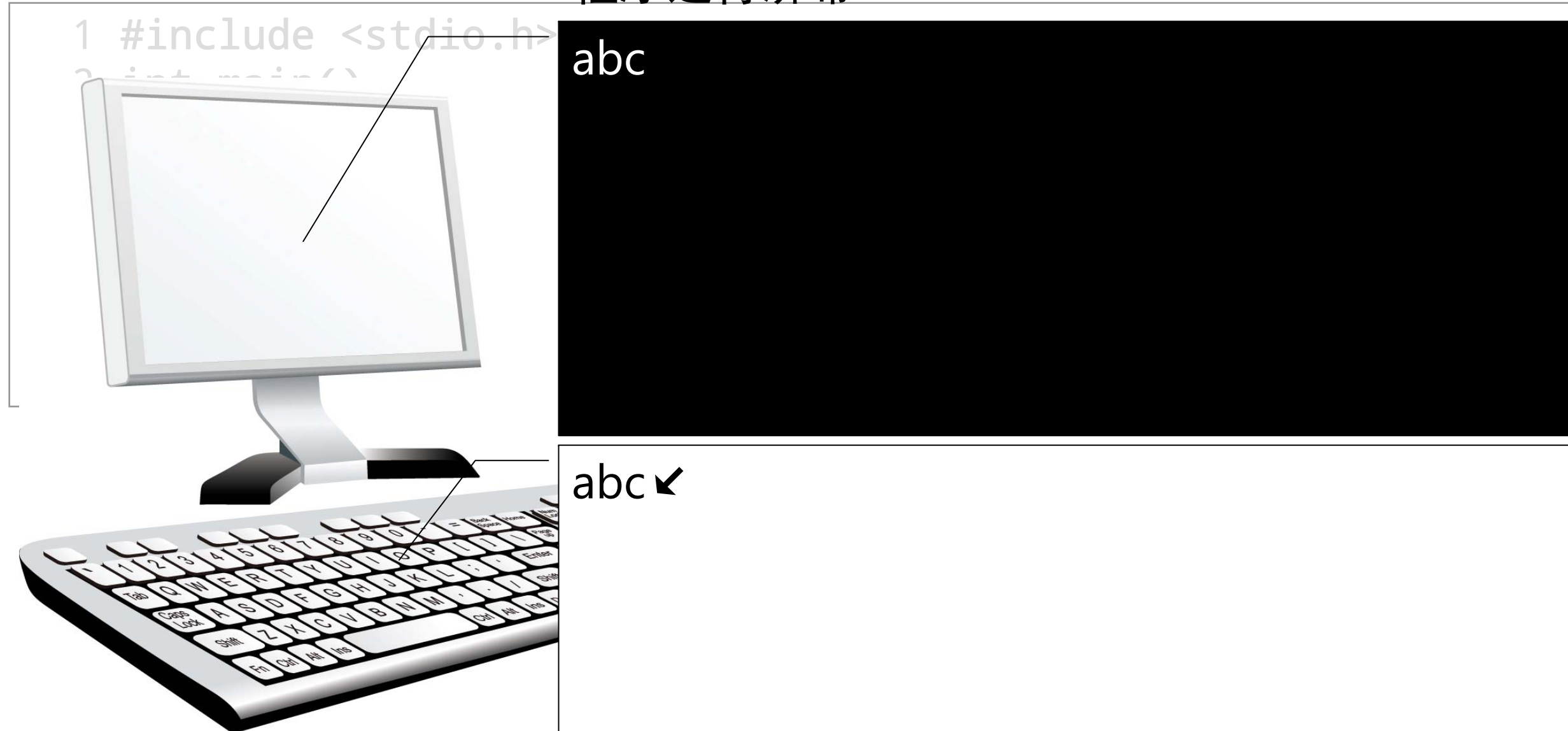
例3.2

```
1 #include <stdio.h>
2 int main()
3 {
4     char c1, c2, c3;
5     c1=getchar(); c2=getchar(); c3=getchar(); //输入字符
6     putchar(c1); putchar(c2); putchar(c3); //输出字符
7     return 0;
8 }
```

3.2.1 字符输入与输出

例3.2

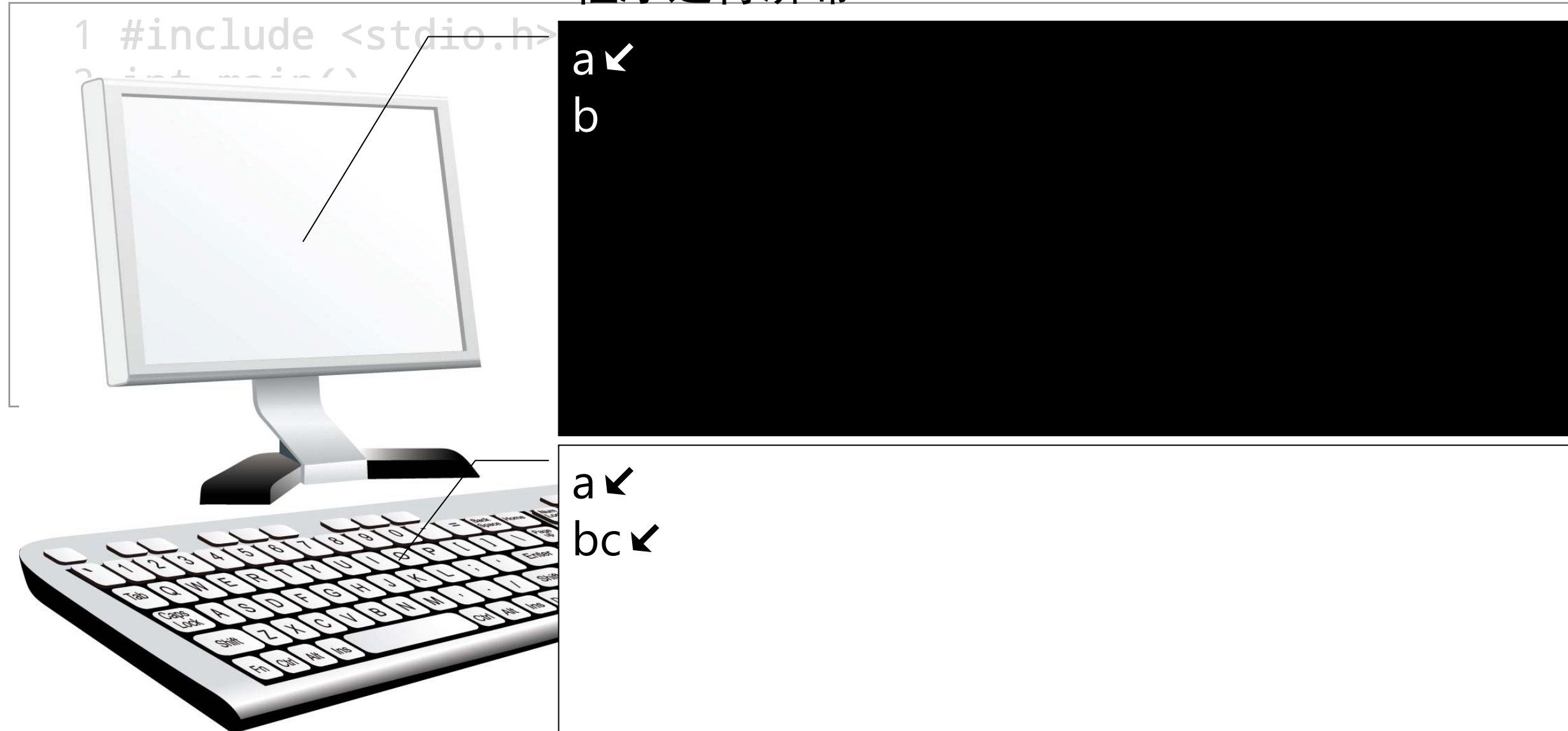
程序运行屏幕



3.2.1 字符输入与输出

例3.2

程序运行屏幕



CP 程序设计