

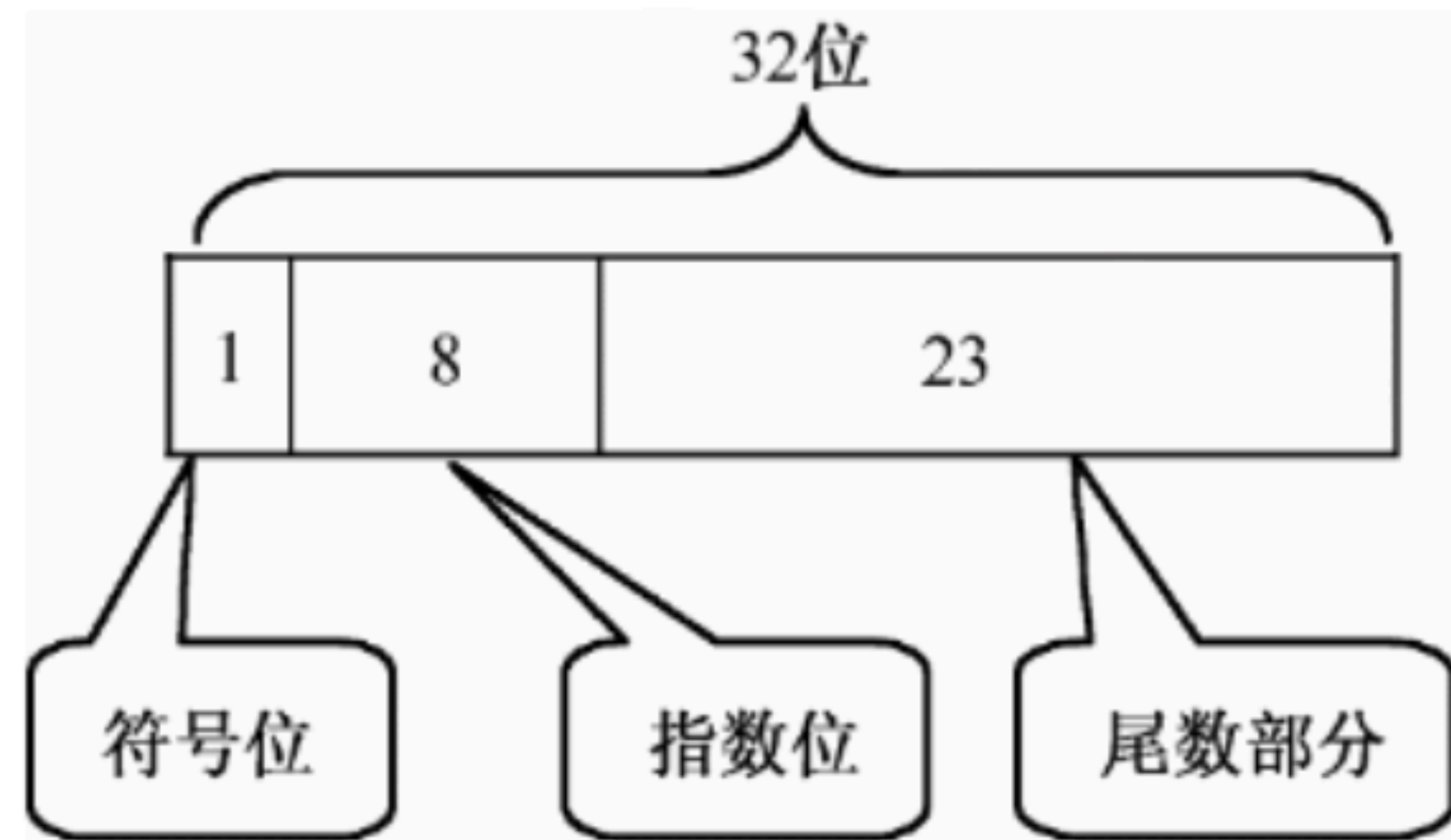


# 知识点总结

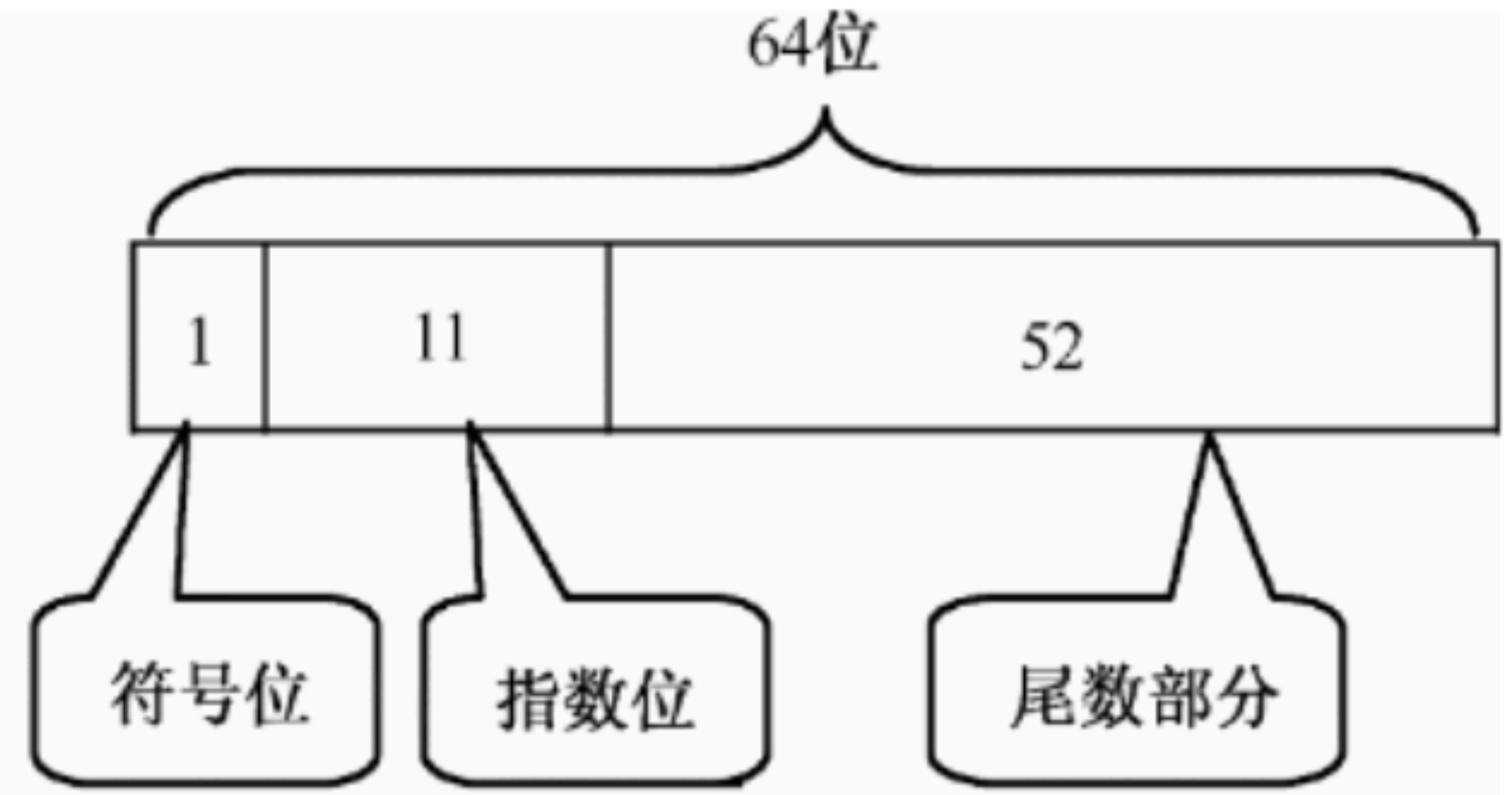


- **double类型**在内存中占8个字节，能表示范围约为 $-1.79 \times 10^{308} \sim 1.79 \times 10^{308}$ ，通常可表示15~16位十进制有效数字位数。
- 定义一个double类型变量例子：`double double_1 = 10.0;`  
打印一个double类型变量例子：`printf("%lf", double_1);`（默认保留6位小数）
- 无论是float类型还是double类型，在内存中都分为3个部分：
  - 1) **符号位**：0表示正，1表示负。
  - 2) **指数位**：用于储存科学计数法中的指数部分。
  - 3) **尾数部分**：用于储存尾数部分（即有效数字）。

尾数部分占的位数越多，数的有效数字越多，精度越高。指数部分占的位数越多，则能表示的数值范围越大。



float类型



double类型



- 由于浮点数类型的表示位数有限，float类型和double类型在储存和运算中都可能存在舍入误差。
- 对于int、float、double类型变量，如果赋值超出其可表示的范围，会导致数据溢出（即无法正确表示该数值）。
- **char类型**在内存中占1个字节，最多只能表示256（即 $2^8$ ）种不同的字符。
- char类型变量只能储存**单个字符**，比如说一个字母、数字或其它符号。
- 定义一个char类型变量例子：`char char_1 = 'A'`；（注意：字符用单引号包裹！）。

打印一个char类型变量例子：`printf("%c", char_1);`

- Char类型实际上储存的是字符对应的整数值，字符和整数值之间的对应关系是由**ASCII码**定义的。





· 标准ASCII码总共定义了128个字符和对应整数值。比如，根据ASCII码表，大写字母F对应的整数值是70。

ASCII value	Character	Control character	ASCII value	Character	ASCII value	Character	ASCII value	Character
000	(null)	NUL	032	(space)	064	@	096	
001	☺	SOH	033	!	065	A	097	a
002	☹	STX	034	"	066	B	098	b
003	♥	ETX	035	#	067	C	099	c
004	♦	EOT	036	\$	068	D	100	d
005	♠	ENQ	037	%	069	E	101	e
006	▲	ACK	038	&	070	F	102	f
007	[beep]	BEL	039	'	071	G	103	g
008	■	BS	040	(	072	H	104	h
009	(tab)	HT	041	)	073	I	105	i
010	(line feed)	LF	042	*	074	J	106	j
011	(home)	VT	043	+	075	K	107	k
012	(form feed)	FF	044	,	076	L	108	l
013	(carriage return)	CR	045	-	077	M	109	m
014	♪	SO	046	.	078	N	110	n
015	☼	SI	047	/	079	O	111	o
016	▶	DLE	048	0	080	P	112	p
017	◀	DC1	049	1	081	Q	113	q
018	↑	DC2	050	2	082	R	114	r
019	!!	DC3	051	3	083	S	115	s
020	π	DC4	052	4	084	T	116	t
021	§	NAK	053	5	085	U	117	u
022	▬	SYN	054	6	086	V	118	v
023	↕	ETB	055	7	087	W	119	w
024	↕	CAN	056	8	088	X	120	x
025	↓	EM	057	9	089	Y	121	y
026	→	SUB	058	:	090	Z	122	z
027	←	ESC	059	;	091	[	123	{
028	(cursor right)	FS	060	<	092	\	124	}
029	(cursor left)	GS	061	=	093	]	125	~
030	(cursor up)	RS	062	>	094	^	126	-
031	(cursor down)	US	063	?	095	_	127	☐

Copyright 1989, JazPac Cam. Copyright 1987, Loading Edge Computer Products, Inc.



- 将ASCII表上的十进制整数值转换为八进制或十六进制后，可以使用相应的**转义符**来表示对应的字符。

- **八进制的转义符**：以'\ '开头，后面跟着最多三位的八进制数字，例如'\010'；

- **十六进制的转义符**：以'\x'开头，后面跟着16进制数，例如'\x4b'。

例如，大写字母F的ASCII码值是70，可以用'\106'（八进制表示）或'\x46'（十六进制表示）表示。

- C语言允许对int类型变量赋予字符值，也允许对char类型变量赋整数值。在输出时，允许把char类型变量按整数形式输出，也允许把int类型变量按字符输出。

例子：

```
int int_1 = 65;
```

```
int int_2 = 'A'; // 'A'会被当作对应的ASCII码65赋给变量int_2
```

```
char char_1 = 'A';
```

```
char char_2 = 65; // 将ASCII码65对应字符'A'，赋给变量char_2
```

```
printf("%c %d", int_1, char_1); // 将int类型变量按字符形式输出，  
char类型变量按整数形式输出
```



- 可以同时定义多个相同类型的变量，各变量名之间用逗号间隔。

例子：`int a, b, c;`

- 对多个变量赋初始值时，有以下几种方式：

1) 先定义，再分别赋初始值。

例子：

```
int a, b, c;
```

```
a=2;
```

```
b=8;
```

2) 定义的同时，给部分变量赋初始值。

例子：`int a=2, b, c;`

- 即使给多个变量赋同样的初始值，也要分别赋值，不能用等号连接。

```
int a=6, b=6, c=6; (√)
```

```
int a=b=c=6; (×)
```



- 占位符:

指定打印时保留n位小数: **%.nf**

指定打印的数值总宽度为n位: **%nf**【数值原本的宽度（包括整数、小数点和小数部分）不足时，会填充空格；宽度超出时，会完整打印数值，不会截断】

- 可以用**sizeof**来获取数据类型或变量所占的字节数。

例子: `#include <stdio.h>`

```
int main() {  
    // 输出 int 类型所占的字节数  
    printf("int 类型的字节数: %zu\n", sizeof(int));  
    // 输出 char 类型所占的字节数  
    printf("char 类型的字节数: %zu\n", sizeof(char));  
    // 输出 float 类型所占的字节数  
    printf("float 类型的字节数: %zu\n", sizeof(float));  
    // 输出 double 类型所占的字节数  
    printf("double 类型的字节数: %zu\n", sizeof(double));  
    return 0;  
}
```





在整数型的有效范围内，二进制可以精确表示任意一个整数，没有误差，而对实数型进行十进制转二进制时，指数部分和位数部分都有位数限制，通常会出现舍入误差，没办法精确表示。

题目1、关于C语言中数的表示，以下叙述中正确的是（ A ）

- A. 只有整型数在允许范围内能精确无误地表示，实型数会有误差
- B. 只要在允许范围内整型和实型都能精确表示
- C. 只有实型数在允许范围内能精确无误地表示，整型数会有误差
- D. 只有八进制表示的数才不会有误差



题目2、若有定义语句char c='\101'; 则变量C在内存中占 ( **B** )

A. 2个字节

B. 1个字节

C. 3个字节

D. 4个字节



题目3、有以下程序

```
#include<stdio.h>
int main()
{
    int s,t,A=10;
    double B=6;
    s=sizeof(A);
    t=sizeof(B);
    printf("%d,%d\n",s,t);
    return 0;
}
```

在VC++2010平台上编译运行，程序运行后的输出结果是（ A ）

A. 4,8

B. 4,4

C. 2,4

D. 10,6



题目4、有以下定义语句，编译时会出现编译错误的是（ C ）

A. `char a = 'a';`

B. `char a = '\n';` 换行符

C. `char a = 'aa';`

D. `char a = '\x2d';` 十六进制转义符



不同变量之间要用逗号分隔  
给多个变量赋同样的初始值，也要分别赋值

题目5、以下选项中正确的定义语句是（ C ）

A. `double a;b;`

B. `double a=b=7;`

C. `double a=7,b=7;`

D. `double,a,b;`





题目6、以下能正确输出字符a的语句是 ( A )

A. `printf("%c", 'a');`

B. `printf("%f", 'a');`

`%f` -> 浮点数

C. `printf("%c", "a");`

D. `printf("%d", 'a');`

`%d` -> 十进制整数



题目7、有以下程序：

```
#include<stdio.h>
int main()
{
    char ch;
    int k;
    ch='a';
    k =12;
    printf("%c,%d,", ch, ch, k); a,97,
    printf("k=%d\n", k);         k=12
    return 0;
}
```

已知字符a的ASCII码十进制值为97，则执行上述程序段后输出的结果是（ **D** ）

A. 因变量类型与格式描述符的类型不匹配输出无定值

B. 输出项与格式描述符个数不符，输出为零值或不定值

C. a,97,12k=12

D. a,97,k=12



题目8、有以下程序：

```
#include<stdio.h>
int main()
{
    int x=12;
    double y=3.141593;
    printf("%d%8.6f",x,y); 123.141593
    return 0;
}
```

输出结果是 ( A )

A. 123.141593

B. 123.141493

C. 12, 3.141593

D. 123.1415930