


```
show_bytes(valp, 1); /* A. */
show_bytes(valp, 2); /* B. */
show_bytes(valp, 3); /* C. */
```

指出在小端法机器和大端法机器上，每次调用的输出值。


- |         |      |
|---------|------|
| A. 小端法: | 大端法: |
| B. 小端法: | 大端法: |
| C. 小端法: | 大端法: |

 **练习题 2.6** 使用 `show_int` 和 `show_float`，我们确定整数 3510593 的十六进制表示为 0x00359141，而浮点数 3510593.0 的十六进制表示为 0x4A564504。

- 写出这两个十六进制值的二进制表示。
- 移动这两个二进制串的相对位置，使得它们相匹配的位数最多。有多少位相匹配呢？
- 串中的什么部分不相匹配？

#### 2.1.4 表示字符串

C 语言中字符串被编码为一个以 `null` (其值为 0) 字符结尾的字符数组。每个字符都由某个标准编码来表示，最常见的是 ASCII 字符码。因此，如果我们以参数 “12345” 和 6 (包括终止符) 来运行例程 `show_bytes`，我们得到结果 31 32 33 34 35 00。请注意，十进制数字  $x$  的 ASCII 码正好是  $0x3x$ ，而终止字节的十六进制表示为  $0x00$ 。在使用 ASCII 码作为字符码的任何系统上都将得到相同的结果，与字节顺序和字大小规则无关。因而，文本数据比二进制数据具有更强的平台独立性。

 **练习题 2.7** 下面对 `show_bytes` 的调用将输出什么结果？

```
const char *s = "abcdef";
show_bytes((byte_pointer) s, strlen(s));
```

注意字母 ‘a’ ~ ‘z’ 的 ASCII 码为  $0x61 \sim 0x7A$ 。

#### 旁注 文字编码的 Unicode 标准

ASCII 字符集适合于编码英语文档，但是在表达一些特殊字符方面并没有太多办法，例如法语的 “Ç”。它完全不适合编码希腊语、俄语和中文等语言的文档。这些年，提出了很多方法来对不同语言的文字进行编码。Unicode 联合会 (Unicode Consortium) 修订了最全面且广泛接受的文字编码标准。当前的 Unicode 标准 (7.0 版) 的字库包括将近 100 000 个字符，支持广泛的语言种类，包括古埃及和巴比伦的语言。为了保持信用，Unicode 技术委员会否决了为 Klingon (即电视连续剧《星际迷航》中的虚构文明) 编写语言标准的提议。

基本编码，称为 Unicode 的 “统一字符集”，使用 32 位来表示字符。这好像要求文本串中每个字符要占用 4 个字节。不过，可以有一些替代编码，常见的字符只需要 1 个或 2 个字节，而不太常用的字符需要多一些的字节数。特别地，UTF-8 表示将每个字符编码为一个字节序列，这样标准 ASCII 字符还是使用 and 它们在 ASCII 中一样的单字节编码，这也就意味着所有的 ASCII 字节序列用 ASCII 码表示和用 UTF-8 表示是一样的。

Java 编程语言使用 Unicode 来表示字符串。对于 C 语言也有支持 Unicode 的程序库。

#### 2.1.5 表示代码

考虑下面的 C 函数：