《计算机组成原理》

（第二讲习题答案）

厦门大学信息学院软件工程系 曾文华

2023年3月2日

第2章　数据信息的表示

- 2.1　数据表示的作用
- 2.2　数值数据的表示
- 2.3　非数值数据的表示
- 2.4　数据信息的校验

- 例2.1：求补码
- 解：
  - 小数（正数）：x=+0.0101，$[x]_{补}$=0.0101
  - 小数（负数）：x=-0.0101，$[x]_{补}$=2+x=2-0.0101=1.1011
  - 小数（负数）：x=-0.0000，$[x]_{补}$=2+x=2-0.0000=0.0000
  - 小数（负数）：x=-1.0000，$[x]_{补}$=2+x=2-1.0000=1.0000

- 例2.2：设某计算机的字长为8位，分别求真值$x=(-10101)_2$，真值x=-128的补码。
- 解：
  - 整数（负数）：$x=(-10101)_2$，$[x]_{补}$=1 0000 0000+x=1 0000 0000-10101=1110 1011
  - 整数（负数）：x=-128，$[x]_{补}$=256+x=256-128=128=1000 0000

- 例2.3：求补码
- 解：
  - 整数（负数）：x=-1000，$[x]_{补}$=1 0000+x=1 0000-1000=1,1000
  - 整数（负数）：x=-0001，$[x]_{补}$=1 0000+x=1 0000-0001=1,1111
  - 小数（负数）：x=-0.0001，$[x]_{补}$=2+x=2-0.0001=1.1111

- 例2.4：根据补码求真值
- 解：
  - 整数（负数）：$[x]_{补}$=1,1000，x=1 0000-$[x]_{补}$=1 0000-1 1000=-1000
  - 整数（负数）：$[x]_{补}$=1,1111，x=1 0000-$[x]_{补}$=1 0000-1 1111=-0001

- **例2.5：求变形补码**
- **解：**
  - 小数：**x=-0.0101，[x]$_{补}$=4+x=4-0.0101=11.1011**
  - （取反加1：**0101**取反**=1010，**加**1=1011，[x]$_{补}$=11.1011**）

- **例2.6：求移码**
- **解：**
  - **x=+1010110，[x]$_{补}$=0,1010110， [x]$_{移}$=1,1010110**
  - **x=-1010110，[x]$_{补}$=1,0101010， [x]$_{移}$=1,0101010**

- 例2.7：将十进制数20.59375转换成IEEE754单精度浮点数的十六进制机器码。
- 解：
  - 20.59375 = 10100.10011
  - 10100.10011 = 1.010010011×2^4 = $1.M \times 2^e$
  - S=0（正数），e=4，E=e+127=131 = 10000011，M = 010010011
  - 单精度浮点数格式：0　1000 0011　010 0100 1100 0000 0000 0000
  - 最终的机器码=0100 0001 1010 0100 1100 0000 0000 0000=41A4C000H

- 例2.8：求IEEE754单精度浮点数C136000H对应的十进制值。
- 解：
  - C136000H=1100 0001 0011 0110 0000 0000 0000
  - 单精度浮点数=1　1000 0010　011 0110 0000 0000 0000
  - S=1（负数），E=1000 0010=130，M=011011
  - e=E-127=130-127=3
  - 尾数=1.M=1.011011
  - 浮点数对应的十进制值=$-2^3 \times 1.011011$=-1011.011=-11.375

- 例2.9：已知 $f(n) = \sum_{i=0}^{n} 2^i = 2^{n+1}-1 = 11\ldots1B$ (n+1个1)，计算f(n)的C语言函数f1如下：
  - 例如：n=0，f(0)=1；n=1，f(1)=11；n=7，f(7)=1111 1111。

```
int f1(unsigned n)
{
   int sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
       power *= 2;
       sum += power;
   }
   return sum;
}
```

```
float f2(unsigned n)
{
   float sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
       power *= 2;
       sum += power;
   }
   return sum;
}
```

- 将f1中的int型数据都改为float型数据，可得到计算f(n)的另一个函数f2。假设unsigned和int型数据都占32位，float型数据采用IEEE754单精度标准。请回答以下问题：
- （1）当n=0时，f1会出现死循环，为什么？若将f1中的变量i和n都定义为int型，则f1是否还会出现死循环？为什么？
- （2）f1(23)和f2(23)的返回值是否相等？机器数各是什么（用十六进制表示）？
- （3） f1(24)和f2(24)的返回值分别是33554431和33554432.0，为什么不相等？
- （4）f(31)=2^{32}-1，而f1(31)的返回值却为-1，为什么？若要使f1(n)的返回值与f(n)相等，则最大的n是多少？
- （5）f2(127)的机器数为7F80 0000H，对应的值是什么？若要使f2(n)的结果不溢出，则最大的n是多少？

- 例2.9：已知 $f(n)=\sum_{i=0}^{n}2^i = 2^{n+1}-1 = 11\ldots1B$ (n+1个1)，计算f(n)的C语言函数f1如下：
  - 例如：n=0，f(0)=1；n=1，f(1)=11；n=7，f(7)=1111 1111。

```
int f1(unsigned n)
{
   int sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
      power *= 2;
      sum += power;
   }
   return sum;
}
```

```
float f2(unsigned n)
{
   float sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
      power *= 2;
      sum += power;
   }
   return sum;
}
```

- 将f1中的int型数据都改为float型数据，可得到计算f(n)的另一个函数f2。假设unsigned和int型数据都占32位，float型数据采用IEEE754单精度标准。请回答以下问题：
- （1）当n=0时，f1会出现死循环，为什么？若将f1中的变量i和n都定义为int型，则f1是否还会出现死循环？为什么？
- 解：
  - 由于i和n都是unsigned型（32位），n=0时，n-1=0-1=11…11（32个1）$=2^{32}-1$，“i<=n-1”永远成立，因此出现死循环。
  - 如果将i和n都定义为int型，n=0时，n-1=0-1=-1，“i<=n-1”条件不成立，直接退出循环，不会出现死循环。

- 例2.9：已知 $f(n)=\sum_{i=0}^{n} 2^i = 2^{n+1}-1$ = 11…1B (n+1个1)， 计算f(n)的C语言函数**f1**如下：
  - 例如：n=0，f(0)=1；n=1，f(1)=11；n=7，f(7)=1111 1111。

```
int f1(unsigned n)
{
   int sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
       power *= 2;
       sum += power;
   }
   return sum;
}
```

```
float f2(unsigned n)
{
   float sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
       power *= 2;
       sum += power;
   }
   return sum;
}
```

- 将f1中的int型数据都改为float型数据，可得到计算f(n)的另一个函数**f2**。假设unsigned和int型数据都占32位，float型数据采用IEEE754单精度标准。请回答以下问题：
- （2）f1(23)和f2(23)的返回值是否相等？机器数各是什么（用十六进制表示）？
- 解：
  - f(23)=1111 1111 1111 1111 1111 1111B（24个1）=FF FFFFH，该数可用int型数据表示，所以f1(23)=**00FF FFFFH= 16,777,215**
  - 该数也可以转换为float型，表示为：1.111 1111 1111 1111 1111 1111x2^{23}，采用IEEE754标准单精度浮点数表示为：
    - 阶码：E=23+127=150=1001 0110
    - 尾数：M=111 1111 1111 1111 1111 1111
    - 符号位：S=0
  - f2(23)=0 1001 0110 111 1111 1111 1111 1111 1111=**4B7F FFFFH**=2^{E-127}x1.M=2^{23}x($2-2^{-23}$)=$2^{24}-2^{0}$= **16,777,215**
  - f1(23)和f2(23)的返回值相等。f1(23)的机器数=00FF FFFFH；f2(23)的机器数=4B7F FFFFH。

- 例2.9：已知 $f(n)=\sum_{i=0}^{n}2^i$ = 2^{n+1}-1 = 11…1B (n+1个1)，计算f(n)的C语言函数**f1**如下：
  - 例如：n=0，f(0)=1；n=1，f(1)=11；n=7，f(7)=1111 1111。

```
int f1(unsigned n)
{
   int sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
       power *= 2;
       sum += power;
   }
   return sum;
}
```

```
float f2(unsigned n)
{
   float sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
       power *= 2;
       sum += power;
   }
   return sum;
}
```

- 将f1中的int型数据都改为float型数据，可得到计算f(n)的另一个函数**f2**。假设unsigned和int型数据都占32位，float型数据采用IEEE754单精度标准。请回答以下问题：
- （3） f1(24)和f2(24)的返回值分别是33 554 431和33 554 432.0，为什么不相等？
- 解：
  - f(24)=1 1111 1111 1111 1111 1111 1111B（25个1）=1FF FFFFH，该数可用int型数据表示，所以f1(24)=01FF FFFFH=33 554 431
  - 该数也可以转换为float型，表示为：1.1111 1111 1111 1111 1111 1111x2^{24}，采用IEEE754标准单精度浮点数表示为：
    - 阶码：E=24+127=151=1001 0111
    - 尾数：M=000 0000 0000 0000 0000 0000（舍入处理，末位加1）
    - 符号位：S=0
  - f2(24)=0 1001 0111 000 0000 0000 0000 0000 0000=4B80 0000H
  - f2(24)=1.Mx2^{E-127}=1.000 0000 0000 0000 0000 0000x2^{24}=2x2^{24}=2^{25}= 33 554 432
  - 故f1(24)和f2(24)的返回值不相等。原因是浮点数尾数只有23位，f(24)的尾数有24位，舍入处理，末位加1。

- 例2.9：已知 $f(n)=\sum_{i=0}^{n}2^i$ = 2^{n+1}-1 = 11…1B (n+1个1)，计算f(n)的C语言函数f1如下：
  - 例如：n=0，f(0)=1；n=1，f(1)=11；n=7，f(7)=1111 1111。

```
int f1(unsigned n)
{
   int sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
      power *= 2;
      sum += power;
   }
   return sum;
}
```

```
float f2(unsigned n)
{
   float sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
      power *= 2;
      sum += power;
   }
   return sum;
}
```

- 将f1中的int型数据都改为float型数据，可得到计算f(n)的另一个函数f2。假设unsigned和int型数据都占32位，float型数据采用IEEE754单精度标准。请回答以下问题：
- （4）f(31)=2^{32}-1，而f1(31)的返回值却为-1，为什么？若要使f1(n)的返回值与f(n)相等，则最大的n是多少？
- 解：
  - f(31)=11…11（32个1）=FFFF FFFFH= 4,294,967,295= 2^{32}-1
  - f1(31)=FFFF FFFFH=-1（补码）
  - 因为f1的返回值sum是整数（有符号数），有符号数FFFF FFFFH为-1。
  - 若要使f1(n)的返回值与f(n)相等，则n的最大值为30。
  - n=30时，f(30)=011…11（31个1）=7FFF FFFFH= 2,147,483,647；f1(30)=7FFF FFFFH= 2,147,483,647（补码）。

- 例2.9：已知 $f(n)=\sum_{i=0}^{n} 2^i=2^{n+1}-1=11\ldots1B$(n+1个1)，计算f(n)的C语言函数f1如下：
  - 例如：n=0，f(0)=1；n=1，f(1)=11；n=7，f(7)=1111 1111。

```
int f1(unsigned n)
{
   int sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
       power *= 2;
       sum += power;
   }
   return sum;
}
```

```
float f2(unsigned n)
{
   float sum=1,power=1;
   for(unsigned i=0; i<=n-1; i++)
   {
       power *= 2;
       sum += power;
   }
   return sum;
}
```

- 将f1中的int型数据都改为float型数据，可得到计算f(n)的另一个函数f2。假设unsigned和int型数据都占32位，float型数据采用IEEE754单精度标准。请回答以下问题：

- （5）f2(127)的机器数为7F80 0000H，对应的值是什么？若要使f2(n)的结果不溢出，则最大的n是多少？

- 解：
  - 浮点数=7F80 0000H=0111 1111 1000 0000 0000 0000 0000 0000，E=1111 1111=255，M=0，对应值为+∞（见教材表2.6）。
  - 当n=126时，f(126)=1…1(127个1)=2^{127}-1=(2-2^{-126})x2^{126}=1.1…1(126个1)x2^{126}，对应的阶码为E=127+126=253，尾数部分舍入后阶码加1，最终阶码E=254，是单精度浮点数的最大阶码。故要使f2(n)的结果不溢出，最大的n是126。

- 例2.10：现有两种编码体系，分别分析它们各自的码距：
  - （1）设用4位二进制数表示16种状态：0000～1111
  - （2）4位二进制数可表示8种状态：0000、0011、0101、0110、1001、1010、1100、1111

- 解：
  - 第（1）种编码的码距为1（例如0000和0001，只有1位不同），任何一个合法编码发生一位错误时，就会变成另外一位合法编码，因此这种编码不具备检测错误的能力。
  - 第（2）种编码的最小码距为2（例如0000和00011，有2位不同），任何一个合法编码如果发生一位错误时（如0000变成1000），就会变成无效编码，因此这种编码可以识别一位错误。但是发生二位错误时（如0000变成0011），又会变成另一个合法的编码，因此该编码对两位错误无法检测。

- 例2.11：设7位ASCII码= $D_7……D_2D_1$=1101010，请给出能纠正1位错的海明码方案。在假设没有3位错的前提下，尝试分析该编码能否区分1位错和2位错。

- 解：
  - 原始数据=$D_7……D_2D_1$=1101010，k=7，则r=4
  - 校验码$P_4P_3P_2P_1$如下（参见表2.22）：
    - $P_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7 = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1=1$
    - $P_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1=1$
    - $P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1=0$
    - $P_4 = D_5 \oplus D_6 \oplus D_7 = 0 \oplus 1 \oplus 1=0$
  - 海明码$H_{11}…H_2H_1 = D_7D_6D_5P_4D_4D_3D_2P_3D_1P_2P_1$=11001010011
  - 该海明码只能发现并纠正1位错误。假设在传输或存储过程中，该海明码的第6位（H_6）出错，传输或存储后的扩展海明码= $H'_{11}…H'_2H'_1$=11001110011
  - 检错码为：
    - $G_1 = P'_1 \oplus D'_1 \oplus D'_2 \oplus D'_4 \oplus D'_5 \oplus D'_7 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 =0$
    - $G_2 = P'_2 \oplus D'_1 \oplus D'_3 \oplus D'_4 \oplus D'_6 \oplus D'_7 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 =1$
    - $G_3 = P'_3 \oplus D'_2 \oplus D'_3 \oplus D'_4 = 0 \oplus 1 \oplus 1 \oplus 1=1$
    - $G_4 = P'_4 \oplus D'_5 \oplus D'_6 \oplus D'_7 = 0 \oplus 0 \oplus 1 \oplus 1=0$
  - 检错码$G=G_4G_3G_2G_1$=0110=6，表示出现1位错，且是第6位出错，只需将第6位取反即可得到正确的数据。因此，该海明码可以发现并纠正1位错。

- 例2.11：设7位ASCII码= D_7……D_2D_1=1101010，请给出能纠正1位错的海明码方案。在假设没有3位错的前提下，尝试分析该编码能否区分1位错和2位错。

- 解（续）：
  - 但是该海明码不能发现2位出错。
  - 假设在传输或存储过程中，该海明码的第6位（H_6）和第4位（H_4）出错，传输或存储后的扩展海明码= $H'_{11}…H'_2H'_1$=11001111011
  - 检错码为：
    - $G_1 = P'_1 \oplus D'_1 \oplus D'_2 \oplus D'_4 \oplus D'_5 \oplus D'_7 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 0$
    - $G_2 = P'_2 \oplus D'_1 \oplus D'_3 \oplus D'_4 \oplus D'_6 \oplus D'_7 = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$
    - $G_3 = P'_3 \oplus D'_2 \oplus D'_3 \oplus D'_4 = 1 \oplus 1 \oplus 1 \oplus 1 = 0$
    - $G_4 = P'_4 \oplus D'_5 \oplus D'_6 \oplus D'_7 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$
  - 检错码$G=G_4G_3G_2G_1$=0010=2，显然G的值反映不了海明码是不是2位出错。
  - 如果要能发现2位错，就需要使用扩展的海明码。

- 例2.12：模2乘法运算
- 解：
  - 1101 x 101 = 111001

```
   1101
  x 101
  -----
   1101
  0000
+1101
-------
 111001
```

- 例2.13：模2除法运算
- 解：
  - 10010 ÷ 101 = 商为101、余数为11

```
       101
     ------
101 | 10010
     -101
      ---
       011
      -000
       ---
        110
       -101
        ---
         11
```

- 例2.14：求CRC码
- 解：
  - 原始数据=110，k=3，r=4，n=7；M(x)·2^r=110 0000
  - 选择生成多项式G(x)=11101
  - M(x)·2^r/G(x) = 110 0000 / 11101 = 商为101、余数为1001
  - 则CRC码：110 1001

```
                101
        -----------
11101 | 1100000
        -11101
         -----
          01010
         -00000
          -----
           10100
          -11101
           -----
            1001
```

- （补充）例1：设原始数据=1010，求该数据的海明码。假设在传输或存储过程中，该海明码出现1位错误，请验证海明码能够发现并纠正1位错。

- 解：
  - 原始数据=$D_4D_3D_2D_1$=1010，k=4，则r=3
  - 校验码$P_3P_2P_1$如下：
    - $P_1 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1=0$
    - $P_2 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1=1$
    - $P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1=0$
  - 海明码$H_7 \ldots H_2H_1 = D_4D_3D_2P_3D_1P_2P_1$=1010010
  - 假设在传输或存储过程中，海明码的第7位（H_7）出错，传输或存储后的海明码=$H'_7 \ldots H'_2H'_1=D'_4D'_3D'_2P'_3D'_1P'_2P'_1$=0010010
  - 检错码为：
    - $G_1 = P'_1 \oplus D'_1 \oplus D'_2 \oplus D'_4 = 0 \oplus 0 \oplus 1 \oplus 0=1$
    - $G_2 = P'_2 \oplus D'_1 \oplus D'_3 \oplus D'_4 = 1 \oplus 0 \oplus 0 \oplus 0=1$
    - $G_3 = P'_3 \oplus D'_2 \oplus D'_3 \oplus D'_4 = 0 \oplus 1 \oplus 0 \oplus 0=1$
  - $G_3G_2G_1$=111，表示H_7出错，只需要将该位取反，即可得到正确的海明码，说明海明码能够发现并纠正1位错。

- （补充）例2：假设接收到的海明码为1010110，且假设接收到的海明码最多出现1位错误。请问该海明码对应的原始数据是多少？

- 解：
  - 接收到的海明码$H'_7 \ldots H'_2 H'_1 = D'_4 D'_3 D'_2 P'_3 D'_1 P'_2 P'_1 = 1010110$
  - 检错码为：
    - $G_1 = P'_1 \oplus D'_1 \oplus D'_2 \oplus D'_4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$
    - $G_2 = P'_2 \oplus D'_1 \oplus D'_3 \oplus D'_4 = 1 \oplus 1 \oplus 0 \oplus 1 = 1$
    - $G_3 = P'_3 \oplus D'_2 \oplus D'_3 \oplus D'_4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$
  - $G_3 G_2 G_1 = 011$，表示H'_3出错，只需要将该位取反，即可得到正确的海明码
  - 正确的海明码$= H_7 \ldots H_2 H_1 = D_4 D_3 D_2 P_3 D_1 P_2 P_1 = 1010010$
  - 原始数据$= D_4 D_3 D_2 D_1 = 1010$

- （补充）例3：设原始数据=1010，求该数据的扩展海明码。假设在传输或存储过程中，该扩展海明码分别出现：无错、1位错误、2位错误，请验证扩展海明码能够发现2位错误并纠正1位错。
- 解：
  - 原始数据=$D_4D_3D_2D_1$=1010，k=4，则r=3
  - 校验码$P_3P_2P_1$如下：
    - $P_1 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1=0$
    - $P_2 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1=1$
    - $P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1=0$
  - 海明码$H_7…H_2H_1 = D_4D_3D_2P_3D_1P_2P_1$=1010010
  - 总偶校验码$P_{all} = (D_1 \oplus D_2 \oplus D_3 \oplus D_4) \oplus (P_1 \oplus P_2 \oplus P_3)$=1
  - 扩展海明码= $H_7…H_2H_1P_{all}$ =10100101
  - （1）假设在传输或存储过程中，扩展海明码没有出现错误，传输或存储后的扩展海明码= $H'_7…H'_2H'_1P'_{all}=D'_4D'_3D'_2P'_3D'_1P'_2P'_1P'_{all}$=10100101
  - 检错码为：
    - $G_1 = P'_1 \oplus D'_1 \oplus D'_2 \oplus D'_4 = 0 \oplus 0 \oplus 1 \oplus 1=0$
    - $G_2 = P'_2 \oplus D'_1 \oplus D'_3 \oplus D'_4 = 1 \oplus 0 \oplus 0 \oplus 1=0$
    - $G_3 = P'_3 \oplus D'_2 \oplus D'_3 \oplus D'_4 = 0 \oplus 1 \oplus 0 \oplus 1=0$
  - 总偶检错码$G_{all} = P'_{all} \oplus (D'_1 \oplus D'_2 \oplus D'_3 \oplus D'_4) \oplus (P'_1 \oplus P'_2 \oplus P'_3)=1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0=0$
  - G_{all}=0，$G=G_3G_2G_1$=000，表示没有出错。

- （补充）例3：设原始数据=1010，求该数据的扩展海明码。假设在传输或存储过程中，该扩展海明码分别出现：无错、1位错误、2位错误，请验证扩展海明码能够发现2位错误并纠正1位错。
- 解（续）：
  - 原始数据=$D_4D_3D_2D_1$=1010，k=4，则r=3
  - 校验码$P_3P_2P_1$如下：
    - $P_1 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1=0$
    - $P_2 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1=1$
    - $P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1=0$
  - 海明码$H_7 \ldots H_2H_1 = D_4D_3D_2P_3D_1P_2P_1$=1010010
  - 总偶校验码$P_{all} = (D_1 \oplus D_2 \oplus D_3 \oplus D_4) \oplus (P_1 \oplus P_2 \oplus P_3)=1$
  - 扩展海明码= $H_7 \ldots H_2H_1P_{all}$ =10100101

  - （2-1）假设在传输或存储过程中，扩展海明码的第8位（H_7）出错，传输或存储后的扩展海明码= $H'_7 \ldots H'_2H'_1P'_{all}=D'_4D'_3D'_2P'_3D'_1P'_2P'_1P'_{all}$=00100101

  - 检错码为：
    - $G_1 = P'_1 \oplus D'_1 \oplus D'_2 \oplus D'_4 = 0 \oplus 0 \oplus 1 \oplus 0=1$
    - $G_2 = P'_2 \oplus D'_1 \oplus D'_3 \oplus D'_4 = 1 \oplus 0 \oplus 0 \oplus 0=1$
    - $G_3 = P'_3 \oplus D'_2 \oplus D'_3 \oplus D'_4 = 0 \oplus 1 \oplus 0 \oplus 0=1$

  - 总偶检错码$G_{all} = P'_{all} \oplus (D'_1 \oplus D'_2 \oplus D'_3 \oplus D'_4) \oplus (P'_1 \oplus P'_2 \oplus P'_3)=1 \oplus 0 \oplus 1 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0=1$

  - G_{all}=1，$G=G_3G_2G_1=111 \neq 0$，表示出现1位错，可以根据检错码G的值纠错；$G=G_3G_2G_1=111$，表示H_7出错，只需要将该位取反，即可得到正确的海明码，说明扩展海明码能够发现并纠正1位错。

- （补充）例3：设原始数据=1010，求该数据的扩展海明码。假设在传输或存储过程中，该扩展海明码分别出现：无错、1位错误、2位错误，请验证扩展海明码能够发现2位错误并纠正1位错。
- 解（续）：
  - 原始数据=$D_4D_3D_2D_1$=1010，k=4，则r=3
  - 校验码$P_3P_2P_1$如下：
    - $P_1 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1=0$
    - $P_2 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1=1$
    - $P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1=0$
  - 海明码$H_7 \ldots H_2H_1 = D_4D_3D_2P_3D_1P_2P_1$=1010010
  - 总偶校验码$P_{all} = (D_1 \oplus D_2 \oplus D_3 \oplus D_4) \oplus (P_1 \oplus P_2 \oplus P_3)$=1
  - 扩展海明码= $H_7 \ldots H_2H_1P_{all}$ =10100101
  - （2-2）假设在传输或存储过程中，扩展海明码的第1位（P_{all}）出错，传输或存储后的扩展海明码= $H'_7 \ldots H'_2H'_1P'_{all} = D'_4D'_3D'_2P'_3D'_1P'_2P'_1P'_{all}$=10100100
  - 检错码为：
    - $G_1 = P'_1 \oplus D'_1 \oplus D'_2 \oplus D'_4 = 0 \oplus 0 \oplus 1 \oplus 1=0$
    - $G_2 = P'_2 \oplus D'_1 \oplus D'_3 \oplus D'_4 = 1 \oplus 0 \oplus 0 \oplus 1=0$
    - $G_3 = P'_3 \oplus D'_2 \oplus D'_3 \oplus D'_4 = 0 \oplus 1 \oplus 0 \oplus 1=0$
  - 总偶检错码$G_{all} = P'_{all} \oplus (D'_1 \oplus D'_2 \oplus D'_3 \oplus D'_4) \oplus (P'_1 \oplus P'_2 \oplus P'_3)=0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0 \oplus 1 \oplus 0=1$
  - G_{all}=1，$G=G_3G_2G_1$=000，表示出现1位错，并且是P_{all}出错，不需要纠错。

- （补充）例3：设原始数据=1010，求该数据的扩展海明码。假设在传输或存储过程中，该扩展海明码分别出现：无错、1位错误、2位错误，请验证扩展海明码能够发现2位错误并纠正1位错。
- 解（续）：
  - 原始数据=$D_4D_3D_2D_1$=1010，k=4，则r=3
  - 校验码$P_3P_2P_1$如下：
    - $P_1 = D_1 \oplus D_2 \oplus D_4 = 0 \oplus 1 \oplus 1 = 0$
    - $P_2 = D_1 \oplus D_3 \oplus D_4 = 0 \oplus 0 \oplus 1 = 1$
    - $P_3 = D_2 \oplus D_3 \oplus D_4 = 1 \oplus 0 \oplus 1 = 0$
  - 海明码$H_7 \ldots H_2H_1 = D_4D_3D_2P_3D_1P_2P_1$=1010010
  - 总偶校验码$P_{all} = (D_1 \oplus D_2 \oplus D_3 \oplus D_4) \oplus (P_1 \oplus P_2 \oplus P_3)$=1
  - 扩展海明码= $H_7 \ldots H_2H_1P_{all}$ =10100101
  - （3）假设在传输或存储过程中，扩展海明码的第8位（H_7）和第6位（H_5）出错，传输或存储后的扩展海明码= $H'_7 \ldots H'_2H'_1P'_{all} = D'_4D'_3D'_2P'_3D'_1P'_2P'_1P'_{all}$=00000101
  - 检错码为：
    - $G_1 = P'_1 \oplus D'_1 \oplus D'_2 \oplus D'_4 = 0 \oplus 0 \oplus 0 \oplus 0 = 0$
    - $G_2 = P'_2 \oplus D'_1 \oplus D'_3 \oplus D'_4 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$
    - $G_3 = P'_3 \oplus D'_2 \oplus D'_3 \oplus D'_4 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$
  - 总偶检错码$G_{all} = P'_{all} \oplus (D'_1 \oplus D'_2 \oplus \ldots\ldots \oplus D'_k) \oplus (P'_1 \oplus P'_2 \oplus \ldots\ldots \oplus P'_r) = 1 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0$
  - $G_{all}=0$，$G=G_3G_2G_1=110 \neq 0$，表示出现2位错，但是无法知道是哪2位错误，即无法纠错。

习题（P53-56）

- 2.2
- 2.4
- 2.5
- 2.6
- 2.7
- 2.9
- 2.10
- 2.13
- 2.16
- 2.17
- 2.18

习题 2

2.1　解释下列名词。

真值　机器码　原码　反码　补码　移码　模　定点数　浮点数　溢出　精度溢出　浮点数规格化　隐藏位　BCD 码　有权码　无权码　BID 码　DPD 码　二进制浮点数　十进制浮点数　ASCII 码　机内码　字形码　字库　码距　校验码　多重奇偶校验　ECC 码　海明码　CRC 码

2.2　选择题（考研真题）。

（1）[2015] 由 3 个 “1” 和 5 个 “0” 组成的 8 位二进制补码，能表示的最小整数是________。

A. −126　　B. −125　　C. −32　　D. −3

（2）[2019] 考虑以下 C 语言代码：

```
unsigned short usi=65535;
short si=usi;
```

执行上述程序段后，si 的值是 ________。

A. −1　　B. −32767　　C. −32768　　D. −65535

（3）[2012] 假定编译器规定 int 和 short 类型长度分别为 32 位和 16 位，执行下列 C 语言语句：unsigned short x=65530；unsigned int y=x；得到 y 的机器数为 ______。

A. 0000 7FFAH　　B. 0000 FFFAH

C. FFFF 7FFAH　　D. FFFF FFFAH

（4）[2016] 有如下 C 语言程序段：short si=−32767；unsigned short usi=si；执行上述两条语句后，usi 的值为 ______。

A. −32767　　B. 32767　　C. 32768　　D. 32769

（5）[2011]float 型数据通常用 IEEE754 单精度浮点数格式表示。若编译器将 float 型变量 x 分配在一个 32 位浮点寄存器 FR1 中，且 x=−8.25，则 FR1 的内容是 ______。

A. C104 0000H　　B. C242 0000H

C. C184 0000H　　D. C1C2 0000H

（6）[2013] 某数采用 IEEE754 单精度浮点数格式表示为 C640 0000H，则该数的值是 ______。

A. -1.5×2^{13}　　B. -1.5×2^{12}　　C. -0.5×2^{13}　　D. -0.5×2^{12}

（7）[2012]float 型（即 IEEE754 单精度浮点数格式）能表示的最大正整数是 ______。

A. $2^{126}-2^{103}$　　B. $2^{127}-2^{104}$　　C. $2^{127}-2^{103}$　　D. $2^{128}-2^{104}$

（8）[2018]IEEE754 单精度浮点格式表示的数中，最小规格化正数是 ______。

A. 1.0×2^{-126}　　B. 1.0×2^{-127}　　C. 1.0×2^{-128}　　D. 1.0×2^{-149}

（9）[2014]float 型数据通常用 IEEE754 单精度浮点格式表示。假定两个 float 型变量 x 和 y 分别存放在 32 位寄存器 f1 和 f2 中，若 (f1)=CC90 0000H，(f2)=B0C0 0000H，则 x 和 y 之间的关系为 ______。

A. $x<y$ 且符号相同　　B. $x<y$ 且符号不同

C. $x>y$ 且符号相同　　D. $x>y$ 且符号不同

（10）[2010] 假定变量 i、f、d 的数据类型分别为 int、float、double（int 用补码表示，float 和 double 用 IEEE754 标准中的单精度和双精度浮点数据格式表示），已知 i=785，f=1.5678e3，d=1.5e100，若在 32 位计算机中执行下列关系表达式，则结果为真的是 ______。

Ⅰ. i==(int)(float)i　　Ⅱ. f==(float)(int)f　　Ⅲ. f==(float)(double)f　　Ⅳ. (d+f)−d==f

A. 仅Ⅰ、Ⅱ　　B. 仅Ⅰ、Ⅲ　　C. 仅Ⅱ、Ⅲ　　D. 仅Ⅲ、Ⅳ

（11）[2013] 用海明码对长度为 8 位的数据进行检错和纠错时，若能纠正一位错，则校验位数至少为 ______。

A. 2　　B. 3　　C. 4　　D. 5

2.3　回答下列问题。

（1）为什么计算机中采用二进制进行数据表示和运算？

（2）相对于奇偶校验，交叉奇偶校验的检错与纠错能力的提高需要付出哪些方面的代价？

（3）为什么计算机中采用补码表示带符号的整数？

（4）浮点数的表示范围和精度分别由什么决定？

（5）汉字输入码、机内码和字形码在汉字处理过程中各有何作用？

（6）在机内码中如何区分 ASCII 字符和汉字字符？

（7）为什么现代处理器中又开始支持十进制浮点数运算？

（8）如何识别浮点数的正负？浮点数能表示的数值范围和数值的精度取决于什么？

（9）浮点数有两个 0 会带来什么问题？

（10）简述 CRC 校验码的检错原理，CRC 能纠错吗？

2.4 写出下列各数的原码、反码和补码。

0，−0，0.10101，−0.10101，0.11111，−0.11111，−0.10000，0.10000

2.5 已知数的补码表示形式，求数的真值。

$[x]_补=0.10010$，$[x]_补=1.10010$，$[x]_补=1.11111$，

$[x]_补=1.00000$，$[x]_补=0.10001$，$[x]_补=1.00001$。

2.6 C 语言中允许无符号数和有符号整数之间的转换，下面是一段 C 语言代码。

```
int x =-1;
unsigned u=2147483648;
printf ("x=%u=%d\n", x, x);
printf ("u=%u=%d\n", u, u);
```

给出在 32 位计算机中上述程序段的输出结果并分析原因。

2.7 分析下列几种情况下所能表示的数据范围分别是多少。

（1）16 位无符号数；

（2）16 位原码定点小数；

（3）16 位补码定点小数；

（4）16 位补码定点整数。

2.8 用补码表示二进制整数，机器码为 $x_0x_1x_2x_3x_4x_5x_6x_7$，x_0 为符号位，补码的模为多少？

2.9 用 IEEE754 32 位单精度浮点数标准表示下列十进制数。

（1）$-6\frac{5}{8}$；（2）3.1415927；（3）64000。

2.10 求与单精度浮点数 43940000H 对应的十进制数。

2.11 求单精度浮点数能表示的最大数和最小数。

2.12 设有两个正浮点数：$N_1=2^m\times M_1$，$N_2=2^n\times M_2$。

（1）若 $m<n$，是否有 $N_1>N_2$？

（2）若 M_1 和 M_2 是规格化的数，上述结论是否正确？

2.13 设二进制浮点数的阶码为 3 位，尾数为 7 位。用模 2 补码写出它们所能表示的最大正数、最小正数、最大负数和最小负数，并将它们转换成十进制数。

2.14 将下列十进制数表示成浮点规格化数，阶码为 4 位，尾数为 10 位，各含 1 位符号，阶码和尾数均用补码表示。

（1）57/128；（2）−69/128。

2.15 设有效信息为 01011011，分别写出其奇校验码和偶校验码。如果接收方收到的有效信息为 01011010，说明如何发现错误。

2.16 由 6 个字符的 7 位 ASCII 字符排列，再加上水平和垂直偶校验位构成表 2.27 所示的行列结构（最后一列 HP 为水平奇偶校验位，最后一行 VP 为垂直奇偶校验位）。

表 2.27 ASCII 交叉校验

字符	7 位 ASCII 字符							HP
3	0	X_1	X_2	0	0	1	1	0
Y_1	1	0	0	1	0	0	X_3	1
+	X_4	1	0	1	0	1	1	0
Y_2	0	1	X_5	X_6	1	1	1	1
D	1	0	0	X_7	1	0	X_8	0
=	0	X_9	1	1	1	X_{10}	1	1
VP	0	0	1	1	1	X_{11}	1	X_{12}

则 X_1、X_2、X_3、X_4 处的比特分别为 ____；X_5、X_6、X_7、X_8 处的比特分别为 ____；X_9、X_{10}、X_{11}、X_{12} 处的比特分别为 ____；Y_1 和 Y_2 处的字符分别为 ____ 和 ____。

2.17　设 8 位有效信息为 01101110，试写出它的海明校验码。给出过程，说明分组检测方式，并给出指错字及其逻辑表达式。如果接收方收到的有效信息变成 01101111，说明如何定位错误并纠正错误。

2.18　设要采用 CRC 码传送数据信息 x=1001，当生成多项式为 $G(x)$=1101 时，请写出它的循环冗余校验码。若接收方收到的数据信息为 x'=1101，说明如何定位错误并纠正错误。

实践训练

（1）在 Logisim 中设计包含 16 位数据位的海明码编解码电路，要求能够在假设没有 3 位错的前提下检测出两位错并纠正一位错。

（2）利用组合逻辑电路在 Logisim 中设计一个包含 16 位数据位的并行 CRC 编、解码电路，要求能够在假设没有 3 位错的前提下检测出两位错并纠正一位错。

习题答案（P53-56）

- **2.1：解释下列名词：**
  1. 真值：+1010，-0.1101
  2. 机器码：由符号和数值一起编码表示的二进制数称为机器码（或机器数）。
  3. 原码：正数的原码数值位为本身，符号位为0；负数的原码数值位为本身，符号位为1。
  4. 反码：正数的反码数值位为本身，符号位为0；负数的反码数值位为本身取反，符号位为1。
  5. 补码：正数的补码数值位为本身，符号位为0；负数的补码数值位为本身取反加1，符号位为1。
  6. 移码：整数的移码为补码的符号位取反，移码的符号位为0时表示负数，移码的符号位为1时表示正数，小数没有移码。
  7. 模：也称为模数，例如4位二进制数整数的模为16，二进制小数的模为2。
  8. 定点数：定点数的小数点是固定的；定点小数，小数点在符号位的右边；定点整数，小数点在最右边。
  9. 浮点数：浮点数的小数点位置是不固定的，小数点位置可以浮动，故称为浮点数。
  10. 溢出：包括正上溢、负上溢、正下溢、负下溢。
  11. 精度溢出：对于小数还存在精度的问题，所有不在数轴上的小数都超出了定点小数所能表示的精度，无法表示，此时定点小数发生精度溢出，只能采用舍入的方法近视表示。
  12. 浮点数规格化：尾数为纯小数，且尾数的绝对值大于等于0.5、小于1，即尾数的最高有效位为1（尾数用原码表示）。非规格化的浮点数可以通过左规或右规的方法变为规格化的浮点数。
  13. 隐藏位：当尾数采用原码表示时，规格化的尾数最高有效位一定是1，可以将最高有效位的1隐藏，从而节省1位存储空间，被隐藏的这一位称为隐藏位。

14. BCD码：Binary Coded Decimal，二进制编码的十进制数

15. 有权码：十进制数的4位二进制数码的每一位都有确定的权值。8421码是一种有权码。

16. 无权码：十进制数的4位二进制数码的每一位没有确定的权值。余3码是一种无权码。

17. BID码：Binary Integer Decimal，十进制整数的二进制表示

18. DPD码：Densely Packed Decimal，紧凑十进制编码

19. 二进制浮点数：浮点数的阶码和尾数都用二进制数表示，且阶码为移码，尾数为小数。

20. 十进制浮点数：浮点数的基数为10，不是2；尾数是定点整数，不是定点小数。

21. ASCII码：American Standard Code for Information Interchange，美国信息交换标准代码，用7位二进制数表示128个字符。

22. 机内码：汉字机内码是计算机内部存储、处理加工和传输汉字时所用的统一编码（如国标码、区位码、Unicode编码等）。

23. 字形码：字形码是汉字的输出码，也称字型码。

24. 字库：也称汉字库，存放汉字字形码的字库。

25. 码距：码距（又称海明距离），两个编码对应二进制位不同的个数称为码距。

26. 校验码：校验码是具有发现错误或纠正错误能力的数据编码。

27. 多重奇偶校验：多重奇偶校验：将原始数据分成若干个校验组，每个数据位至少位于两个或两个以上的校验组，当某个数据位出错时，能在多个检错位中被指出（能够改变多个检错位），从而可以知道是哪一位数据位出错。

28. ECC码：海明码本质上是一种多重奇偶校验码，它是一种既能检错也能纠错的校验码（Error-Correcting Codes， ECC）。

29. 海明码：假设原始数据为：Dk…D2D1，共k位；校验位为：Pr…P2P1，共r位；则海明校验码为：Hn…H2H1，共n位，n=k+r，也称(n,k)码

30. CRC码：循环冗余校验（Cyclic Redundancy Check，CRC）是一种基于模2运算的校验码

- **2.2 选择题**

（1）[2015] 由 3 个 “1” 和 5 个 “0” 组成的 8 位二进制补码，能表示的最小整数是________。

A. −126　　B. −125　　C. −32　　D. −3

答案：B：（-126=1000 0010；-125=0101 1001；-32=1110 0000；-3=1111 1101）

（2）[2019] 考虑以下 C 语言代码：

```
unsigned short usi=65535;
short si=usi;
```

执行上述程序段后，si 的值是 _______。

A. −1　　B. −32767　　C. −32768　　D. −65535

答案：A：（si=usi=65535=FFFFH=-1）

（3）[2012] 假定编译器规定 int 和 short 类型长度分别为 32 位和 16 位，执行下列 C 语言语句：unsigned short x=65530；unsigned int y=x；得到 *y* 的机器数为 _______。

A. 0000 7FFAH　　B. 0000 FFFAH

C. FFFF 7FFAH　　D. FFFF FFFAH

答案：B：（unsigned short x=65530=FFFAH，unsigned int y = 0000 FFFAH）

- ## 2.2 选择题

（4）[2016] 有如下 C 语言程序段：short si=−32767；unsigned short usi=si；执行上述两条语句后，usi 的值为 ________。

A. −32767　　B. 32767　　C. 32768　　D. 32769

答案：D：（short si=-32767=1000,0000,0000,0001B（补码）=8001H，unsigned short usi = 8001H=32769）

（5）[2011]float 型数据通常用 IEEE754 单精度浮点数格式表示。若编译器将 float 型变量 x 分配在一个 32 位浮点寄存器 FR1 中，且 x=−8.25，则 FR1 的内容是 ________。

A. C104 0000H　　B. C242 0000H

C. C184 0000H　　D. C1C2 0000H

答案：A：（x=-8.25=-1000.01=1.00001x2^3，E=127+3=130= 10000010，M=00001，单精度浮点数=1 100 0001 0 000 0100 0000 0000 0000 0000 = C104 0000H）

（6）[2013] 某数采用 IEEE754 单精度浮点数格式表示为 C640 0000H，则该数的值是 ________。

A. -1.5×2^{13}　　B. -1.5×2^{12}　　C. -0.5×2^{13}　　D. -0.5×2^{12}

答案：A：（C640 0000H=1100 0110 0100 0000 0000 0000 0000 0000，符号位=1，E=1000 1100=140，M=100 0000 0000 0000 0000 0000，e=E-127=13，单精度浮点数=-$(1.1)_2$x2^{13} = -1.5x2^{13}）

- ## 2.2 选择题

（7）[2012]float 型（即 IEEE754 单精度浮点数格式）能表示的最大正整数是 ______。

A. $2^{126}-2^{103}$　　B. $2^{127}-2^{104}$　　C. $2^{127}-2^{103}$　　D. $2^{128}-2^{104}$

答案：D：（单精度浮点数能表示的最大正数：E=254，M=11111111111111111111111（23个1），单精度浮点数=2^{E-127}x1.M=2^{127}x($2-2^{-23}$) = $2^{128}-2^{104}$；教材表2.6和表2.7）

（8）[2018]IEEE754 单精度浮点格式表示的数中，最小规格化正数是 ______。

A. 1.0×2^{-126}　　B. 1.0×2^{-127}　　C. 1.0×2^{-128}　　D. 1.0×2^{-149}

答案：A：（单精度浮点数最小规格化正数：E=1，M=00000000000000000000000（23个0），单精度浮点数=2^{E-127}x1.M=2^{-126}x$(1.0)_2$ = 1.0x2^{-126}；教材表2.6和表2.7）

（9）[2014]float 型数据通常用 IEEE754 单精度浮点格式表示。假定两个 float 型变量 x 和 y 分别存放在 32 位寄存器 f1 和 f2 中，若 (f1)=CC90 0000H，(f2)=B0C0 0000H，则 x 和 y 之间的关系为 ______。

A. $x<y$ 且符号相同　　B. $x<y$ 且符号不同

C. $x>y$ 且符号相同　　D. $x>y$ 且符号不同

答案：A：（x=(f1)=CC90 0000H=1100 1100 1001 0000 0000 0000 0000 0000，符号位=1，E=1001 1001=153，M=001 0000 0000 0000 0000 0000，e=E-127=26，单精度浮点数x=-$(1.001)_2$x2^{26}=-1.125x2^{13}；y=(f2)=B0C0 0000H=1011 0000 1100 0000 0000 0000 0000 0000，符号位=1，E=0110 0001=97，M=100 0000 0000 0000 0000 0000，e=E-127=-30，单精度浮点数y=-$(1.1)_2$x2^{-30}=-1.5x2^{-30}；故x<y，且符号相同）

- **2.2 选择题**

（10）[2010] 假定变量 i、f、d 的数据类型分别为 int、float、double（int 用补码表示，float 和 double 用 IEEE754 标准中的单精度和双精度浮点数据格式表示），已知 i=785，f=1.5678e3，d=1.5e100，若在 32 位计算机中执行下列关系表达式，则结果为真的是 ______。

Ⅰ. i==(int)(float)i　　Ⅱ. f==(float)(int)f　　Ⅲ. f==(float)(double)f　　Ⅳ. (d+f)-d==f

A. 仅Ⅰ、Ⅱ　　B. 仅Ⅰ、Ⅲ　　C. 仅Ⅱ、Ⅲ　　D. 仅Ⅲ、Ⅳ

答案：B：（I：整数转浮点再转回整数，没有问题；II：浮点转整数再转回浮点，可能有问题；III：单精度浮点转双精度浮点再转回单精度浮点，没有问题；IV：大数+小数-大数，可能有问题）

（11）[2013] 用海明码对长度为 8 位的数据进行检错和纠错时，若能纠正一位错，则校验位数至少为 ______。

A. 2　　B. 3　　C. 4　　D. 5

答案：C：（海明码：$k+r \le 2^r-1$；$k=8$，$r=4$；$8+4 \le 2^4-1$）

- **2.3 回答下列问题：**

① 为什么计算机中采用二进制进行数据表示和运算？

- 答：二进制由于数码最少、容易与简单的物理状态对应、算术逻辑运算电路更容易实现等优势，成为现代计算机中数据表示的不二之选，采用二进制可以表示任何数据信息。

② 相对于奇偶校验，交叉奇偶校验的检错与纠错能力的提高需要付出哪些方面的代价？

- 答：交叉奇偶校验增加了校验位，例如4个7位二进制数，简单的奇偶校验只需要4位校验位（每一个1位），而交叉奇偶校验需要12位校验位。

③ 为什么计算机中采用补码表示带符号的整数？

- 答：0的补码是唯一的，同样长度的二进制数整数，补码的表示范围比原码多一个（如4位二进制数，原码整数表示范围为-7～+7，补码整数表示范围为-8～+7）；采用补码进行运算时，符号位可以直接参与运算，运算时符号位的进位作为模会被自动舍弃，因此补码的减法运算可以转换成加法运算，大大方便了二进制运算。

④ 浮点数的表示范围和精度分别由什么决定？

- 答：浮点数的表示范围由阶码决定、精度由尾数决定。

⑤ 汉字输入码、机内码和字形码在汉字处理过程中各有何作用？

- 答：汉字输入码就是使用英文键盘输入汉字时的编码；汉字机内码是计算机内部存储、处理加工和传输汉字时所用的统一编码（如国标码、区位码、Unicode编码等）；汉字字形码是汉字的输出码，也称字型码。

⑥ 在机内码中如何区分ASCII字符和汉字字符？

- 答：ASCII码的最高有效位是0，汉字字符的最高有效位是1。

⑦ 为什么现代处理器中又开始支持十进制浮点数运算？

- 答：二进制浮点数的最大问题是不能精确表示十进制数（例如$(0.7)_{10}= (0.1011001100110011……)_2$），不精确的二进制浮点数表示会给运算带来很多误差问题，因此在财务金额运算中是不允许采用二进制浮点数的，必须使用十进制浮点数的表示和运算。

⑧ 如何识别浮点数的正负？浮点数能表示的数值范围和数值的精度取决于什么？

- 答：浮点数的第一位为符号位，该位是0表示正数，该位是1表示负数；浮点数的表示范围由阶码决定、精度由尾数决定。

⑨ 浮点数有两个0会带来什么问题？

- 答：IEEE754单精度浮点数有两个机器0，+0和-0；两个机器0使浮点数少了一个表示的值。

⑩ 简述CRC校验码的检错原理，CRC能纠错吗？

- 答：CRC编码：原始数据左移r位，然后与生成多项式相除（模2运算），得到余数，将原始数据和余数拼接起来就是CRC码。CRC解码：将接收到的CRC码与生成多项式相除（模2运算），得到余数，如果余数=0，则没有错误；如果余数不等于0，表示出错，并且根据余数的值，就可以知道是哪一位出错。因此CRC码可以发现并纠正错误。

- **2.4 写出下列各数的原码、反码和补码：**

2.4 写出下列各数的原码、反码和补码。

0，−0，0.10101，−0.10101，0.11111，−0.11111，−0.10000，0.10000

- **答：**

真值	原码	反码	补码
0	0.00…0	0.00…0	0.00…0
−0	1.00…0	1.11…1	0.00…0
0.10101	0.10101	0.10101	0.10101
−0.10101	1.10101	1.01010	1.01011
0.11111	0.11111	0.11111	0.11111
−0.11111	1.11111	1.00000	1.00001
−0.10000	1.10000	1.01111	1.10000
0.10000	0.10000	0.10000	0.10000

- **2.5 已知数的补码表示形式，求数的真值：**

2.5 已知数的补码表示形式，求数的真值。

$[x]_{补}=0.10010$， $[x]_{补}=1.10010$， $[x]_{补}=1.11111$，

$[x]_{补}=1.00000$， $[x]_{补}=0.10001$， $[x]_{补}=1.00001$。

- **答：真值的有效位=补码的有效位“取反+1”**

补码	真值		补码	真值
$[x]_{补}=0.10010$	x=0.10010		$[x]_{补}=1.00000$	x=-1.00000
$[x]_{补}=1.10010$	x=-0.01110		$[x]_{补}=0.10001$	x=0.10001
$[x]_{补}=1.11111$	x=-0.00001		$[x]_{补}=1.00001$	x=-0.11111

- **2.6 给出在32位计算机中上述程序段的输出结果并分析原因：**

2.6 C 语言中允许无符号数和有符号整数之间的转换，下面是一段 C 语言代码。

```c
int x =-1;
unsigned u=2147483648;
printf ("x=%u=%d\n", x, x);
printf ("u=%u=%d\n", u, u);
```

给出在 32 位计算机中上述程序段的输出结果并分析原因。

- 答：
  - 输出结果如下：

    x= 4294967295= -1；

    u= 2147483648= -2147483648

C:\计算机组成原理（2023年）\实验程序（实验一）\高级语言中的数据类型\C语言

```
x=4294967295=-1
u=2147483648=-2147483648

--------------------------------
Process exited after 0.4045 seconds with return value 0
请按任意键继续. . .
```

  - 因为**x=-1**，对应的补码是**FFFF FFFFH**，无符号数**FFFF FFFFH**对应的真值是**4294967295**
  - 因为**u= 2147483648**，对应的机器码=**8000 0000H**，补码**8000 0000H**对应的真值为**-2147483648**
  -

- **2.7 分析下列几种情况下所能表示的数据范围：**

2.7 分析下列几种情况下所能表示的数据范围分别是多少。

（1）16 位无符号数；

（2）16 位原码定点小数；

（3）16 位补码定点小数；

（4）16 位补码定点整数。

- **答：**

1) 16 位无符号数：0～1111 1111 1111 1111，即 0～$2^{16}-1=65535$
2) 16 位原码定点小数：1.111 1111 1111 1111～0.111 1111 1111 1111，即$-(1-2^{-15})$～$1-2^{-15}$
3) 16 位补码定点小数：1.000 0000 0000 0000～0.111 1111 1111 1111，即-1～$1-2^{-15}$
4) 16 位补码定点整数：1000 0000 0000 0000～0111 1111 1111 1111，即-2^{15}～$2^{15}-1$

- **2.8**

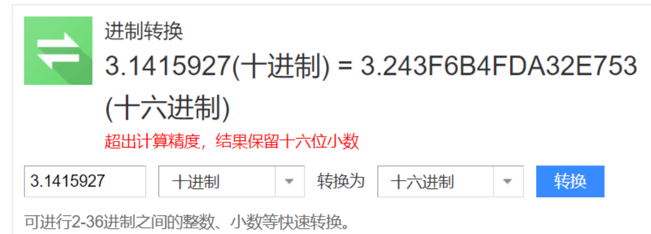
2.8　用补码表示二进制整数，机器码为 $x_0x_1x_2x_3x_4x_5x_6x_7$，x_0 为符号位，补码的模为多少？

- **答：**
  - **1,000 0000 ～0,111 1111 ＝ -128 ～ +127**
  - **机器码为8位，8位二进制数的模为256。**

- **2.9 用IEEE754 32位单精度浮点数标准表示下列十进制数。**

2.9 用 IEEE754 32 位单精度浮点数标准表示下列十进制数。

（1）$-6\frac{5}{8}$；（2）3.1415927；（3）64000。



- **答：参考教材P27-30**
- **（1）-(6+5/8)=$(-110.101)_2$=-1.10101x2^2，E=127+2=129=1000 0001，M=101 0100 0000 0000 0000 0000；单精度浮点数=1 1000 0001 101 0100 0000 0000 0000 0000=C0D40000H**

- **（2）3.1415927=$(3.243F6B)_{16}$=11.0010 0100 0011 1111 0110 1011 =1.1 0010 0100 0011 1111 0110 1011x2^1，E=127+1=128=1000 0000，M=1 0010 0100 0011 1111 0110 11（最后两个11舍入，并进位到上一位）；单精度浮点数=0 1000 0000 1 0010 0100 0011 1111 0110 11=40490FDBH**

- **（3）64000= $(1111101000000000)_2$ = 1.111101000000000 x 2^{15}，E=127+15=142= 1000 1110，M=111 1010 0000 0000 0000 0000；单精度浮点数=0 100 0 111 0 111 1010 0000 0000 0000 0000 = 477A0000H**

- **2.10 求与单精度浮点数43490000H对应的十进制数**

- 答：首先展开成二进制形式，之后按照**IEEE754**标准逐个对应

  - **43940000H=0100 0011 1001 0100 0000 0000 0000 0000，E=10000111=135，M=001 0100 0000 0000 0000 0000，浮点数=1.Mx2^{E-127}=1.00101x2^{8}=1 0010 1000=296**

- **2.11 求单精度浮点数能表示的最大数和最小数。**

- **答：**
  - 单精度浮点数能表示的最大数（最大正数）：**E=254，e=254-127=127，M=11…11，1.M=(2-2^{-23})，fmax = 2^{127}x(2-2^{-23})**
  - 单精度浮点数能表示的最小数（最小负数）：为最大数的负数，**fmin = -fmax = -2^{127}x(2-2^{-23})**
  - 单精度浮点数能表示的最小正数：**E=1，e=1-127=-126，M=00…00，1.M=1，**最小正数**= 2^{-126}x1.0**
  - 单精度浮点数能表示的最大负数：为最小正数的负数，最大负数**= -2^{-126}x1.0**
  -

- **2.12**

2.12 设有两个正浮点数：$N_1=2^m \times M_1$，$N_2=2^n \times M_2$。

（1）若 $m < n$，是否有 $N_1 > N_2$？

（2）若 M_1 和 M_2 是规格化的数，上述结论是否正确？

- 答：
  - （1）有可能，例如，$N_1=2^3x0.1=100=4$；$N_2=2^4x0.001=10=2$；此时$m<n$，但是$N_1>N_2$
  - （2）不可能，因为M_1和M_2是规格化数，则M_1和M_2都在[0.5,1)范围内，即M_1和M_2不可能相差2倍或2倍以上；若$m<n$，则两个数的阶码会相差2倍或2倍以上；因此，若$m<n$，一定有$N_1<N_2$

- **2.13** 设二进制浮点数的阶码为3位，尾数7位。用模2补码写出它们所能表示的最大正数、最小正数、最大负数和最小负数，并将之转换为十进制数。

- 答：

  - 阶码为3位（1位阶符）：-4～3；1,00～0,11
  - 尾数为7位（1位尾符）：-1～+(1-2^{-6})；1.000000～0.111111

  - 最大正数=011 0111111 = 2^3x(1-2^{-6}) = 8-2^{-3}
  - 最小正数=100 0000001 = 2^{-4}x(2^{-6})= 2^{-10}
  - 最大负数=100 1111111 = 2^{-4}x(-2^{-6})= -2^{-10}
  - 最小负数=011 1000000 = 2^3x(-1)= -8

- **2.14**

2.14 将下列十进制数表示成浮点规格化数，阶码为 4 位，尾数为 10 位，各含 1 位符号，阶码和尾数均用补码表示。

（1）57/128；（2）−69/128。

- **答：**
  - **57/128=0.0111001=0.111001x2^{-1}；阶码=-1=1,111；尾数=0.111001000；57/128=1111 0111001000**
  - **-69/128=-0.1000101=-0.1000101x2^{0}；阶码=0=0,000；尾数=1.011101100；-69/128=0000 1011101100**

- **2.15**

2.15 设有效信息为01011011，分别写出其奇校验码和偶校验码。如果接收方收到的有效信息为01011010，说明如何发现错误。

- 答：
  - 原始数据：01011011
  - 奇校验码：01011011 0
  - 偶校验码：01011011 1
  - 接收到的数据= 01011010 （最后一位出错了）
  - 如果是奇校验，则接收到的奇校验码= 01011010 0，此时1的个数是偶数个，说明出错了（假设只有1位出错）
  - 如果是偶校验，则接收到的偶校验码= 01011010 1，此时1的个数是奇数个，说明出错了（假设只有1位出错）

- **2.16**

2.16 由6个字符的7位ASCII字符排列，再加上水平和垂直偶校验位构成表2.27所示的行列结构（最后一列HP为水平奇偶校验位，最后一行VP为垂直奇偶校验位）。

表2.27 ASCII交叉校验

字符	7位ASCII字符							HP
3	0	X_1	X_2	0	0	1	1	0
Y_1	1	0	0	1	0	0	X_3	1
+	X_4	1	0	1	0	1	1	0
Y_2	0	1	X_5	X_6	1	1	1	1
D	1	0	0	X_7	1	0	X_8	0
=	0	X_9	1	1	1	X_{10}	1	1
VP	0	0	1	1	1	X_{11}	1	X_{12}

则X_1、X_2、X_3、X_4处的比特分别为____；X_5、X_6、X_7、X_8处的比特分别为____；X_9、X_{10}、X_{11}、X_{12}处的比特分别为____；Y_1和Y_2处的字符分别为____和____。

表2.27 ASCII交叉校验

字符	7位ASCII码							HP
3	0	X1	X2	0	0	1	1	0
Y1	1	0	0	1	0	0	X3	1
+	X4	1	0	1	0	1	1	0
Y2	0	1	X5	X6	1	1	1	1
D	1	0	0	X7	1	0	X8	0
=	0	X9	1	1	1	X10	1	1
VP	0	0	1	1	1	X11	1	X12

- **2.16**
- **答：**
  - 3的ASCII码=33H=011 0011，因此X1=1、X2=1
  - +的ASCII码=2BH=010 1011，因此X4=0
  - D的ASCII码=44H=100 0100，因此X7=0、X8=0
  - =的ASCII码=3DH=011 1101，因此X9=1、X10=0

字符	7位ASCII码							HP
3	0	X1	X2	0	0	1	1	0
Y1	1	0	0	1	0	0	X3	1
+	X4	1	0	1	0	1	1	0
Y2	0	1	X5	X6	1	1	1	1
D	1	0	0	X7	1	0	X8	0
=	0	X9	1	1	1	X10	1	1
VP	0	0	1	1	1	X11	1	X12

字符	7位ASCII码							HP
3	0	1	1	0	0	1	1	0
Y1	1	0	0	1	0	0	X3	1
+	0	1	0	1	0	1	1	0
Y2	0	1	X5	X6	1	1	1	1
D	1	0	0	0	1	0	0	0
=	0	1	1	1	1	0	1	1
VP	0	0	1	1	1	X11	1	X12

- **2.16**

- **答：**
  - 因为是偶校验，因此X5=1，X6=0，X11=1，X3=1，X12=1
  - Y1=100 1001=49H，为I的ASCII码，Y1=I；
  - Y2=011 0111=37H，为7的ASCII码，Y2=7

字符	7位ASCII码							HP
3	0	X1	X2	0	0	1	1	0
Y1	1	0	0	1	0	0	X3	1
+	X4	1	0	1	0	1	1	0
Y2	0	1	X5	X6	1	1	1	1
D	1	0	0	X7	1	0	X8	0
=	0	X9	1	1	1	X10	1	1
VP	0	0	1	1	1	X11	1	X12

字符	7位ASCII码							HP
3	0	1	1	0	0	1	1	0
Y1	1	0	0	1	0	0	X3	1
+	0	1	0	1	0	1	1	0
Y2	0	1	X5	X6	1	1	1	1
D	1	0	0	0	1	0	0	0
=	0	1	1	1	1	0	1	1
VP	0	0	1	1	1	X11	1	X12

字符	7位ASCII码							HP
3	0	1	1	0	0	1	1	0
I	1	0	0	1	0	0	1	1
+	0	1	0	1	0	1	1	0
7	0	1	1	0	1	1	1	1
D	1	0	0	0	1	0	0	0
=	0	1	1	1	1	0	1	1
VP	0	0	1	1	1	1	1	1

- **2.17 设8位有效信息为01101110，试写出它的海明校验码。给出过程，说明分组检测方式，并给出指错字以及逻辑表达式。如果接收方收到的有效信息变成01101111，说明如何定位错误并纠错。**

- **答：**

  - **原始数据=01101110=D1D2D3D4D5D6D7D8，k=8，r=4，n=k+r=12**

  - **海明码校验位：**
    - **P1=D1⊕D2⊕D4⊕D5⊕D7=0⊕1⊕0⊕1⊕1=1**
    - **P2=D1⊕D3⊕D4⊕D6⊕D7=0⊕1⊕0⊕1⊕1=1**
    - **P3=D2⊕D3⊕D4⊕D8=1⊕1⊕0⊕0=0**
    - **P4=D5⊕D6⊕D7⊕D8=1⊕1⊕1⊕0=1**

  - **海明码=P1P2D1P3D2D3D4P4D5D6D7D8=110011011110** （**校验位P位于 2^n，数据位D顺次排列**）

  - **接收到的数据=01101111（D8出错）**
  - **接收到的海明码=110011011111（D8出错）**
  - **海明码检错位：**
    - **G1=P1⊕D1⊕D2⊕D4⊕D5⊕D7=1⊕0⊕1⊕0⊕1⊕1=0**
    - **G2=P2⊕D1⊕D3⊕D4⊕D6⊕D7=1⊕0⊕1⊕0⊕1⊕1=0**
    - **G3=P3⊕D2⊕D3⊕D4⊕D8=0⊕1⊕1⊕0⊕1=1**
    - **G4=P4⊕D5⊕D6⊕D7⊕D8=1⊕1⊕1⊕1⊕1=1**

  - **G4G3G2G1=1100，表示第12位出错，即D8位出错（从左往右数）**

海明码	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	.....
位置	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	.....
映射关系	P1	P2	D1	P3	D2	D3	D4	P4	D5	D6	D7	D8	D9	D10	D11	.....
G1校验组	√		√		√		√		√		√		√		√	.....
G2校验组		√	√			√	√			√	√			√	√	.....
G3校验组				√	√	√	√					√	√	√	√	.....
G4检验组								√	√	√	√	√	√	√	√	.....
......	......	......	......	......	......	......	......	......	......	......	......	......	......	......	......	.....

每个校验位P负责检测的小组如下：

$P1=D1\oplus D2\oplus D4\oplus D5\oplus D7\oplus D9\oplus D11$

$G1=P1\oplus D1\oplus D2\oplus D4\oplus D5\oplus D7\oplus D9\oplus D11$

$P2=D1\oplus D3\oplus D4\oplus D6\oplus D7\oplus D10\oplus D11$

$G2=P2\oplus D1\oplus D3\oplus D4\oplus D6\oplus D7\oplus D10\oplus D11$

$P3=D2\oplus D3\oplus D4\oplus D8\oplus D9\oplus D10\oplus D11$

$G3=P3\oplus D2\oplus D3\oplus D4\oplus D8\oplus D9\oplus D10\oplus D11$

$P4=D5\oplus D6\oplus D7\oplus D8\oplus D9\oplus D10\oplus D11$

$G4=P4\oplus D5\oplus D6\oplus D7\oplus D8\oplus D9\oplus D10\oplus D11$

- **2.18**

2.18 设要采用 CRC 码传送数据信息 x=1001，当生成多项式为 $G(x)$=1101 时，请写出它的循环冗余校验码。若接收方收到的数据信息为 x'=1101，说明如何定位错误并纠正错误。

- **答：**
  - 原始数据=1001
  - 生成多项式=1101，r=3
  - 将原始数据左移r位得：1001 000
  - 用1001 000 除以生成多项式1101（模2运算），得到余数=011
  - CRC码=1001 011
  - 接收到的数据=1101
  - 接收到的CRC码=1101 011
  - 用1101 011 除以生成多项式1101（模2运算），得到余数=011
  - 根据余数011，查表知道是第2位出错（从左往右数）

1111
1101 | 1001000
-1101
1000
-1101
1010
-1101
1110
-1101
011

1000
1101 | 1101011
-1101
0000
-0000
0001
-0000
0011
-0000
011

Thanks