**Introduction to Computer Science**

**Homework 5**

**沙行勉教授**

**Due Date: 10am, Nov 12, 2018**

**Do not copy others, or you will get punished. If you allow other students to copy your report, you will be punished as well. In the homework, you will get more experience in programming. Good for you. 但是，对于编程题，我希望有基础的人能多帮助其他同学，帮助他们debug, 但是不要让他们抄袭。好吗？**

**PUT DOWN YOUR NAME and EMAIL ADDRESS IN YOUR REPORT.** tThe Filename of your report should be HOMEWORK N\_your name\_your Student ID\_Teacher’s last name (Sha or Zhang), where N is the homework number, for example, *1\_XiaoMing\_2018XXXX\_Sha*. Then send your report to the TA of your class. 注意，两班的TA不一样，千万不要送错了。**你可以用中文来回答问题。**

**请同学阅读书本的第二章和第三章的前面部分。这不是准备高考，你已经是大学生了，要自己学习知识，没人逼你用刷题的方式来学习了。当作有趣的知识来阅读。好吗？**

1. **（5 points）习题2.12， 再加上S3=****A∨(﹁B)∧C**

**A=1,B=0,C=0**

**S1=（1或1与0）=（1与0）=0**

**S2=（1或(1与0)）=（1或0）=1**

**S3=（1或1与0）=（1与0）=0**

1. **（10 points）习题 2.13**

**D:观察发现，当ABC中有奇数个1时，D的值为1，而当ABC中有偶数个1时，D的值则变为0，这与二进制加法的结果一致，将四种D=1的情况列出，分别为**

**A=1，B=0，C=0和A=0，B=1，C=0和A=0，B=0，C=1和A=1，B=1，C=1**

**即ABC和A(﹁B)(﹁C)和B(﹁A)(﹁C)和C(﹁A)(﹁B)**

**E：观察发现，当A=0，B=0，C=1和A=0，B=1，C=0和A=0，B=1，C=1和A=1，B=1，C=1时，E＝1，即ABC和C(﹁A)(﹁B)和B(﹁A)(﹁C)和(﹁A)BC**

**综合：D=ABC+ A(﹁B)(﹁C)+ B(﹁A)(﹁C)+C(﹁A)(﹁B)**

**E=ABC+C(﹁A)(﹁B)+B(﹁A)(﹁C)+(﹁A)BC**

1. **（10 points）习题2.14**
2. **L=BC+BC(A+﹁A)**

**=BC+BC**

**=BC**

1. **L=AE+C(﹁A+﹁E)**

**=AE+C(﹁(AE))**

**=C+AE**

1. **L=AB+(﹁A﹁E)+B(﹁E)**

**=AB+(﹁A﹁E)**

1. **（10 points）习题2.15**

**L=[0,1]**

**for e in L:**

**A=e**

**for e in L:**

**B=e**

**for e in L:**

**C=e**

**D=(A and B and C) or (A and (not B) and (not C)) or ((not A) and B and (not C)) or ((not A) and (not B) and C)**

**if D==True:**

**D=1**

**elif D==False:**

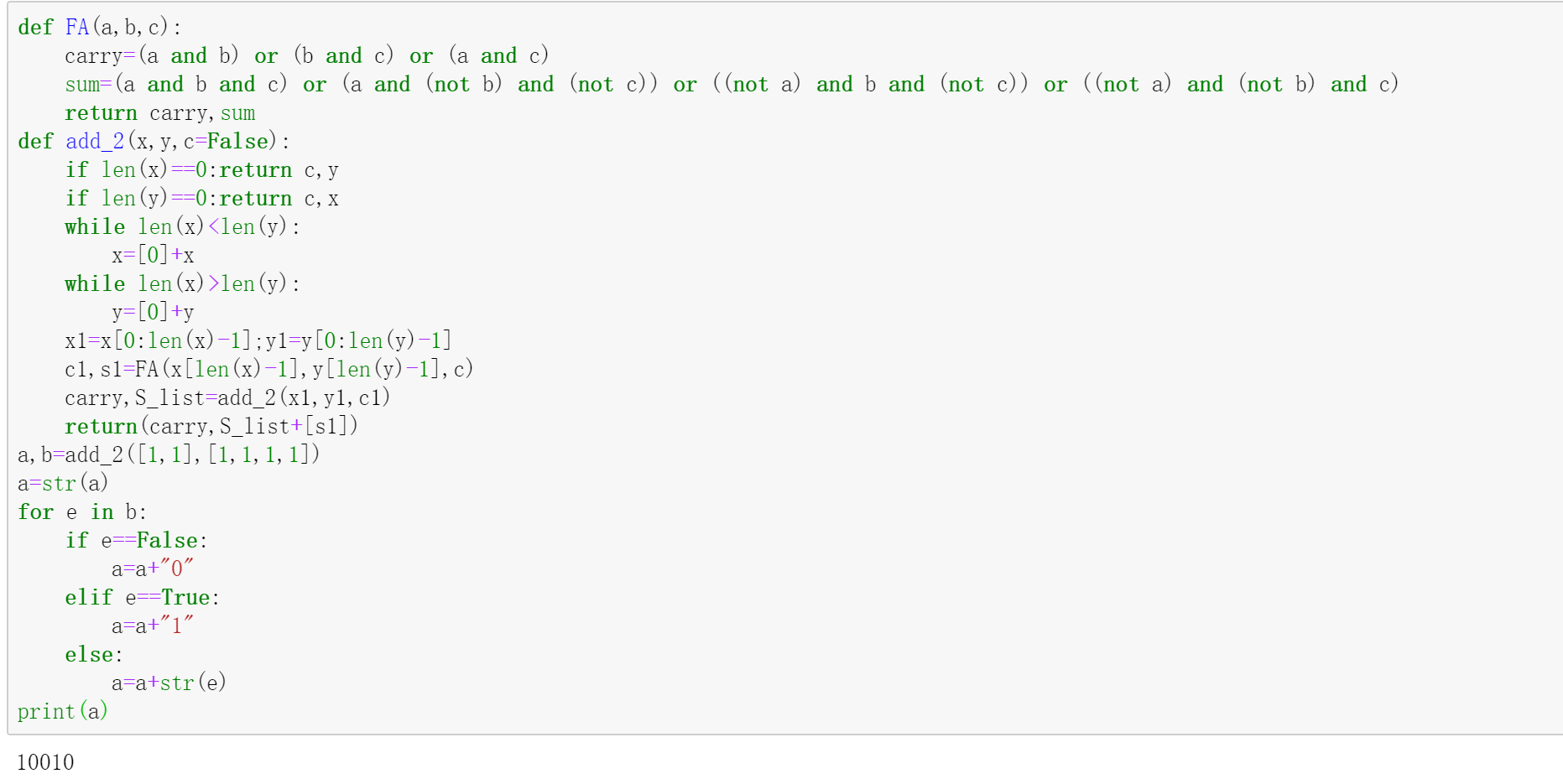
**D=0**

**图片包含 屏幕截图

自动生成的说明 print(A,B,C,D)**

1. **（10 points）习题2.17**

**因为x和y的位数没有确保一致，可能造成空列表与int相加的情况，造成错误，改进代码如下：**

****

1. **（5 points）习题2.19**

**2\*(x^2)-16x+30=0**

**现在的16和30被外星人当成了8进制的数，先将这两个数转成10进制**

**16=8+6=14,30=3\*8+0=24**

**所以x^2-7x+12=0**

**X=3或4**

1. **（10 points）Based on IEEE 754 floating point representation, for 32-bit words, show us how to represent the following floating point numbers: (A) -0.125; (B)1024.875**

**(A)-0.125先不管负号，0.125转化为二进制小数，0.125\*2=0.25,取0,**

**0.25\*2=0.5,取0,0.5\*2=1.0，取1，结束，为0.001，表示为1.0\*(2^-3)**

**因为是负数，第一位为1，-3+127为124，表示二进制为01111100，尾数为23个0，所以表示为10111110000000000000000000000000**

**(B)1024.875:整数部分，1024是2的10次方，所以表示为10000000000，小数部分，0.875\*2=1.75，取1,0.75\*2=1.5，取1,0.5\*2=1，取1结束，0.111，所以为10000000000.111，表示为1.0000000000111\*(2^10),因为是正数，第一位是0,10+127=137，转化为2进制是10001001，尾数是00000000001110000000000，所以为01000100100000000001110000000000**

**8.、（20 points）This programming assignment can be done by you alone or a group of two students. 写下合作同学的名字。**

**输入2个十进制的整数，转为二进制，编写二进制乘法函数，函数返回二进制值，再转为十进制输出。我们现在有正负和位数限制。假设CPU是12位，用补位法表示正负值，则可表示范围是-2048 到 2047 之间的任意值。输入2个十进制的整数，检查介于可表示范围内，就马上转为二进制表示，编写二进制乘法函数mult()，限制在12位，函数返回是否溢出，如果没有溢出返回正确的结果二进制值，再转为十进制输出。**

**例如x=-12, y=100, 最后输出是 -1200。**

**例如 x=-30, y=-10, 最后输出是 300**

**例如，x=4, y=-512, 最后输出是 -2048**

**例如，x=4, y=512, 最后输出是 “溢出错误”**

**例如 x=40, y=-80, 最后输出是 “溢出错误”**

**函数Mult(a, b , num): 输入a, b 是补位后二进制（可以是字符串），num 代表最大限制位数。例如， num=12。 建议如果a, b都是负数，将它们转为正数，再做正数相乘 ；假如a正 b负，则a,b可以交换。让b作为正数，然后一位位看b的位是否0、1，需要时加上移位后的a，如同多个负数相加，只需留下num位部分，仍然能保持它正确的补位值。在函数内检查是否有溢出，绝对不能转为十进制来检查，这不是CPU内使用的方式，在CPU内的运算都是二进制，所以必须在二进制中检查。**

**请同学们把你的代码拷贝到word报告中，并将程序的运行结果截图粘贴到代码 合作同学：董辰尧**

**def base\_2\_add(a,b):#编写一个二进制的加法函数**

**if len(a)<len(b):**

**a,b=b,a**

**a=a[::-1];b=b[::-1]**

**while len(b)<len(a):**

**b=b+"0"**

**i=0;extra=0;L=[]**

**while i<len(a)-1:**

**sum=int(a[i])+int(b[i])+extra**

**if sum==0:**

**r=0;extra=0;L=L+[r]**

**elif sum==1:**

**r=1;extra=0;L=L+[r]**

**elif sum==2:**

**r=0;extra=1;L=L+[r]**

**elif sum==3:**

**r=1;extra=1;L=L+[r]**

**i=i+1**

**if i==len(a)-1:**

**sum=int(a[i])+int(b[i])+extra**

**if sum==0:**

**r=0;L=L+[r]**

**elif sum==1:**

**r=1;L=L+[r]**

**elif sum==2:**

**r=10;L=L+[r]**

**elif sum==3:**

**r=11;L=L+[r]**

**L1=L[::-1];s=""**

**for e in L1:**

**s=s+str(e)**

**return s**

**def Mult(a,b,num):**

**#编写一个将十进制整数转化为二进制函数**

**def base\_ten\_two(x):**

**L=[]**

**s=""**

**x1=abs(x)**

**while x1!=0:**

**r=x1%2;x1=x1//2;L=[r]+L**

**for e in L:**

**s=s+str(e)**

**global num**

**while len(s)<num:**

**s="0"+s**

**return s**

**def multiply\_base2(x,y):**

**i=0;j=len(x)-1**

**a="0"**

**while i<len(x):**

**b=int(x[j])\*y**

**c=str(b)+"0"\*i**

**a=base\_2\_add(a,c)**

**i=i+1**

**j=j-1**

**return (a)**

**if x>(2\*\*(num-1)-1) or x<-(2\*\*(num-1)) or y>(2\*\*(num-1)-1) or y<-(2\*\*(num-1)):**

**print("error")**

**if x>0 and y>0:**

**x1=base\_ten\_two(x)**

**y1=base\_ten\_two(y)**

**if x<0 and y<0:**

**x1=base\_ten\_two(-x)**

**y1=base\_ten\_two(-y)**

**if x\*y<0:**

**x1=base\_ten\_two(abs(x))**

**y1=base\_ten\_two(abs(y))**

**result=int(multiply\_base2(x1,y1))**

**if x\*y>0:**

**if result>=100000000000:**

**return ("溢出错误")**

**elif x\*y<0:**

**if result>100000000000:**

**return ("溢出错误")**

**s\_res=str(result)**

**i=0**

**result\_new=0**

**while i<len(s\_res):**

**result\_new=result\_new+(int(s\_res[i])\*(2\*\*(len(s\_res)-i-1)))**

**i=i+1**

**if x\*y>0:**

**return result\_new**

**if x\*y==0:**

**return 0**

**if x\*y<0:**

**result\_new=result\_new\*(-1)**

**return result\_new**

**x=int(input("请输入一个十进制整数:"))**

**y=int(input("请输入一个十进制整数:"))**

**num=int(input("请输入限制位数:"))**

**Mult(x,y,num)**

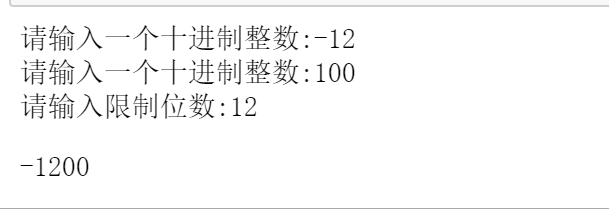
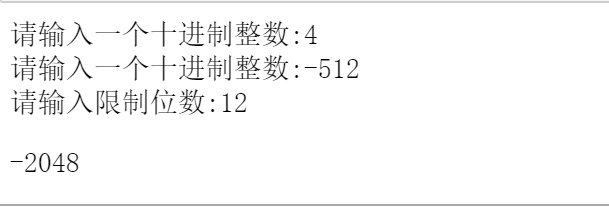
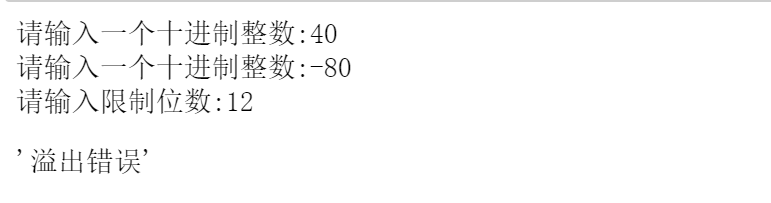
**图片包含 屏幕截图

自动生成的说明**

**图片包含 屏幕截图

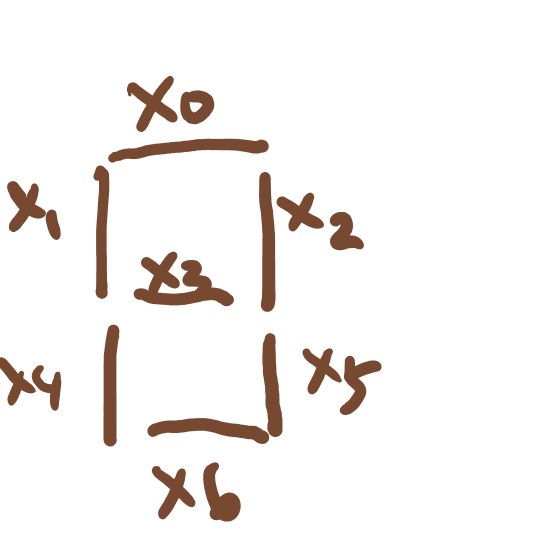
自动生成的说明**

**图片包含 屏幕截图

自动生成的说明图片包含 屏幕截图

自动生成的说明**

**9．（20 points）We are going to design an LED controller that can show decimal numbers 0, 1, 2, 3, 4, 5, 6, 7. Input: 3 signals a, b, c. When (a, b, c)=000, LED will turn on the following bars: x0, x1, x2, x4, x5, x6, which show the shape of number 0. When (a, b, c)=001, LED will turn on x2 and x5, which show the shape of number 1. When (a, b, c)=010, LED will turn on x0, x2, x3, x4, and x6, which show the shape of number 2. Similarly, when (a, b, c)= 011, 100, 101, 110, 111, the corresponding bars will be turned on to show the shapes of numbers 3, 4, 5, 6, 7, respectively. Note that for shape of number 6, x0 is NOT tuned on. You need to show the controlling logic of x0, x1, x2, x3, x4, x5, and x6. (A) Show the truth tables of x0, x1, x2, x3, x4, x5, and x6; (B) Give the Boolean functions using and-or-not logic operations, and try to simplify the logic as much as possible. 请用卡诺图简化，你可以自己到百度查询学习。**



**A B C X0 X1 X2 X3 X4 X5 X6**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | **0** | **0** | **1** | **1** | **1** | **0** | **1** | **1** | **1** |
| **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **1** | **0** | **1** | **1** | **1** | **0** | **1** |
| **0** | **1** | **1** | **1** | **0** | **1** | **1** | **0** | **1** | **1** |
| **1** | **0** | **0** | **0** | **1** | **1** | **1** | **0** | **1** | **0** |
| **1** | **0** | **1** | **1** | **1** | **0** | **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **0** | **1** | **0** | **1** | **1** | **1** | **1** |
| **1** | **1** | **1** | **1** | **0** | **1** | **0** | **0** | **1** | **0** |

**图片包含 文字, 白板

自动生成的说明(B)**

**10．（20 points）This programming assignment can be done by you alone or a group of two students. 写下合作同学的名字。**

**(a) 依据IEEE 754 浮点数表示法格式。写2个函数。假设32位单精准度**

**dec2flost(x): 输入10进制浮点数x（有可能没有小数点，也有可能是负数，用字符串表示）， 输出2进制32位浮点数，以01字符串表示**

**在此，假设尾数23位后就直接舍入，不做进位。假如指数超过2的127次方，视为溢出。小于2的-126次方，可以视为0（全部是0）。请注意对0要特殊处理，我们要将所有位都设为0.**

**第二个函数是转回十进制。函数float2dec(x): 输入是个32位IEEE754 的浮点数，以01字符串表示，返回其十进制数（用字符串表示）。**

**请试验： 1， 1.0， 0， -1000.1， 65535.125, 65535.0125，0.000005， -0.00000000006， 1234567，1234567890， 1.000006， 1.00000006**

**请同学们把你的代码拷贝到word报告中，并将程序的运行结果截图粘贴到代码。请将第一个函数的试验结果打印后，再作为第二个函数的输入。**

**(b) 如果你再完成64位的转换，可以再加5分。**

**合作同学：10185102144，董辰尧**

**def dec2flost(x):**

**if x=='0':return('00000000000000000000000000000000')**

**def convert(a):**

**if a<2:**

**return([a])**

**if a>=2:**

**return(convert(a//2)+[a%2])**

**L=x.split('.')**

**if float(x)>=0:**

**L1=convert(int(L[0]))**

**s1=''**

**for i in L1:**

**s1+=str(i)**

**s1+='.'**

**if float(x)<0:**

**L1=convert(-int(L[0]))**

**s1=''**

**for i in L1:**

**s1+=str(i)**

**s1+='.'**

**s1='-'+s1**

**if len(L)==2:**

**s2='0.'+L[1]**

**c=float(s2)**

**s3=''**

**for i in range(1000):**

**c\*=2**

**if c<1:**

**s3+='0'**

**if c>=1:**

**s3+='1'**

**c-=1**

**s4=s1+s3**

**else:s4=s1+'00000000000000000000000'**

**s5=''**

**if float(s4)<0:**

**s5+='1'**

**else:s5+='0'**

**A=list(s4)**

**A.append('.')**

**for i in range(len(A)):**

**if A[i]=='.':**

**a1=i;break**

**for e in range(len(A)):**

**if A[e]=='1':**

**a2=e;break**

**zhishu=a1-a2-1**

**if zhishu<0:**

**zhishu+=1**

**if zhishu>127:**

**return('溢出')**

**elif zhishu<-126:**

**return('00000000000000000000000000000000')**

**else:**

**B=convert(zhishu+127)**

**if len(B)<8:**

**diff=8-len(B)**

**for y in range(diff):**

**B=[0]+B**

**for o in B:**

**s5+=str(o)**

**for u in range(a2+1,len(A)):**

**if A[u]=='.':continue**

**s5+=A[u]**

**s6=s5[0:32]**

**if len(s6)<32:**

**v=32-len(s6)**

**for j in range (v):**

**s6+='0'**

**return(s6)**

**res1=dec2flost(input("请输入一个十进制数"))**

**print("这个数的IEEE754小数表示为:",res1)**

**def float2dec(x):**

**x1=x[0]**

**if x1=="1":**

**result="-"**

**if x1=="0":**

**result=""**

**x2=x[1:9]**

**x3=x[9:]**

**ex=base\_two\_ten(x2)**

**ex1=ex-127**

**if ex1>0:**

**sz="1"+x3[0:ex1]**

**sf="0."+x3[ex1:]**

**if ex1<0:**

**sz="0"**

**sf="0."+(-1-ex1)\*"0"+"1"+x3**

**if ex1==0:**

**sz="1"**

**sf="0."+(-ex1)\*"0"+x3**

**if sz!="0":**

**sz1=base\_two\_ten(sz)**

**sz2=str(sz1)**

**if sz=="0":**

**sz2="0"**

**sf1=0**

**i=len(sf)-1**

**while i>1:**

**sf1=(sf1+int(sf[i]))/2**

**i=i-1**

**sf2=str(sf1)**

**sf3=sf2[1:]**

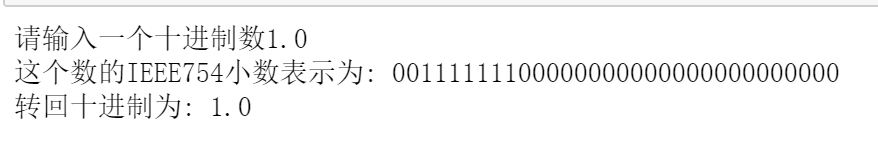
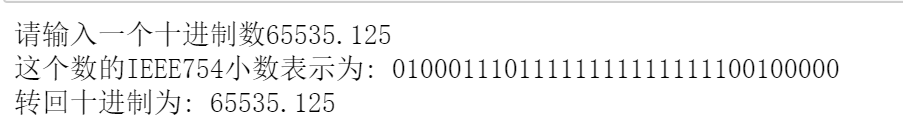
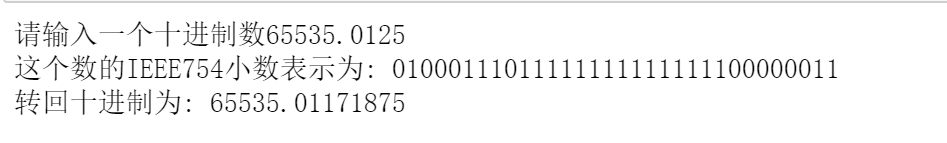
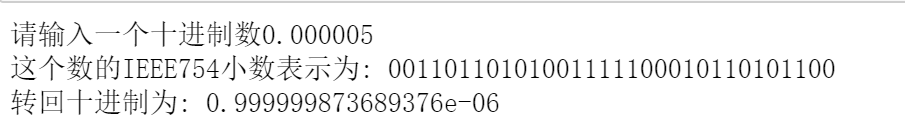
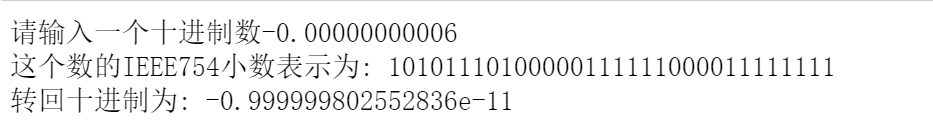
**result=result+sz2+sf3**

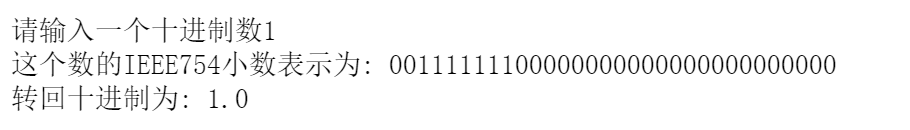
**if x=="00000000000000000000000000000000":**

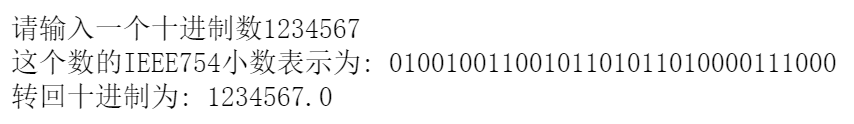
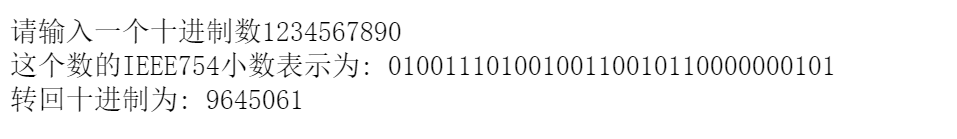
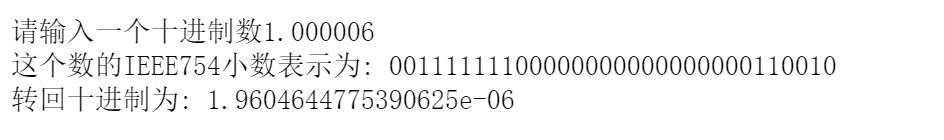
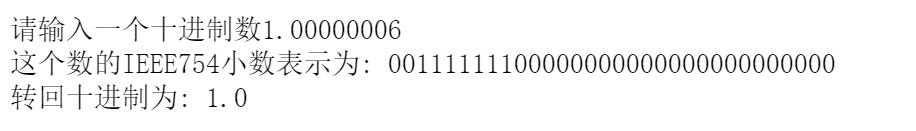
**result="0"**

**return result**

**res2=float2dec(res1)**

**图片包含 屏幕截图

自动生成的说明print("转回十进制为:",res2)**

****