## 一(k=1)

2.3.1

（1）100100010

（2）10001111

（3）11000001001

2.3.4

11111011

2.3.7

移位3，加法3

2.3.11

192

## 二(k=0)

2.5.1

1024b

536.870912GB

2.5.5

0x6c99

0x8001

0x5e08

2.5.7

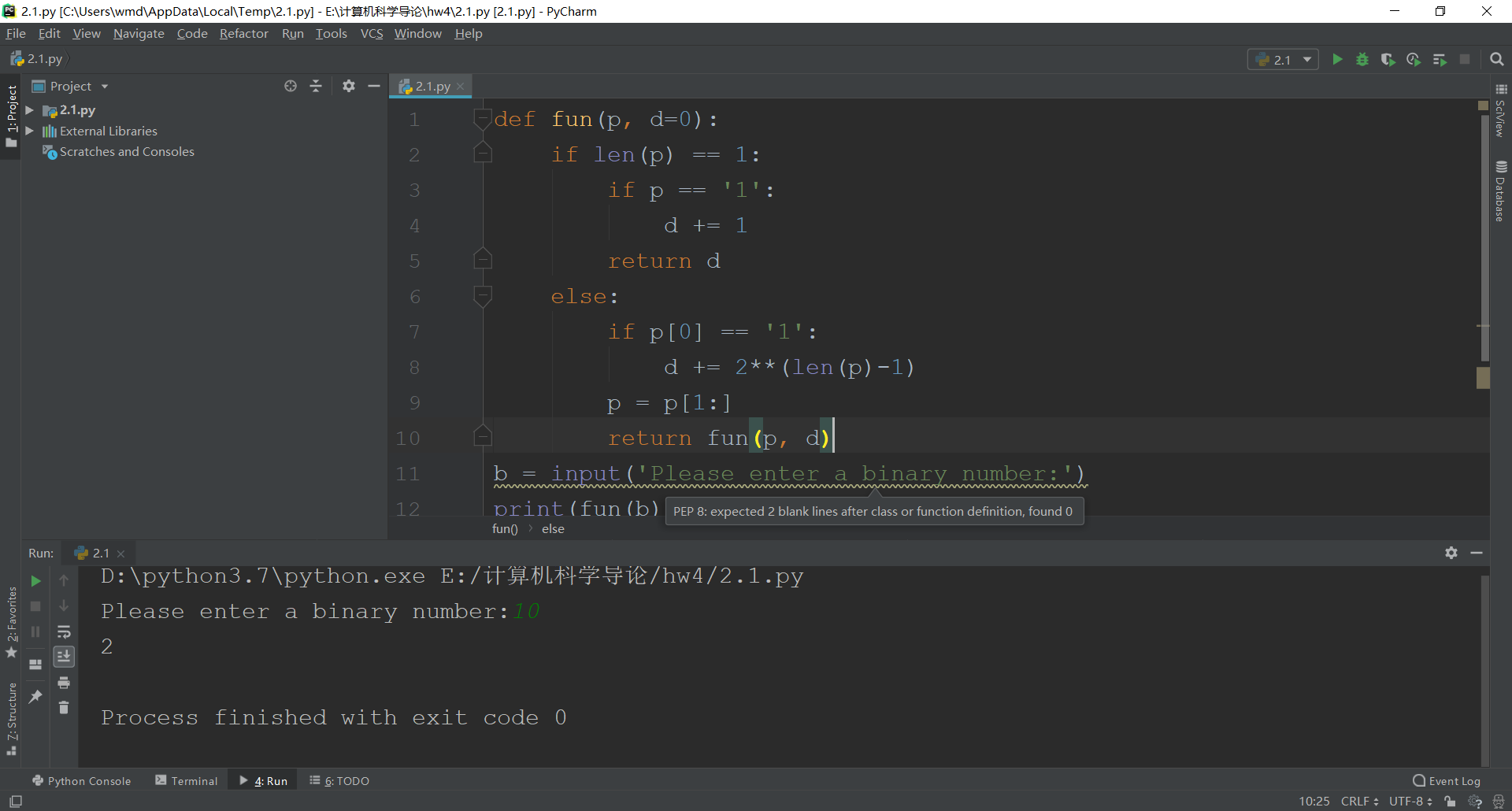
o

?

## 三

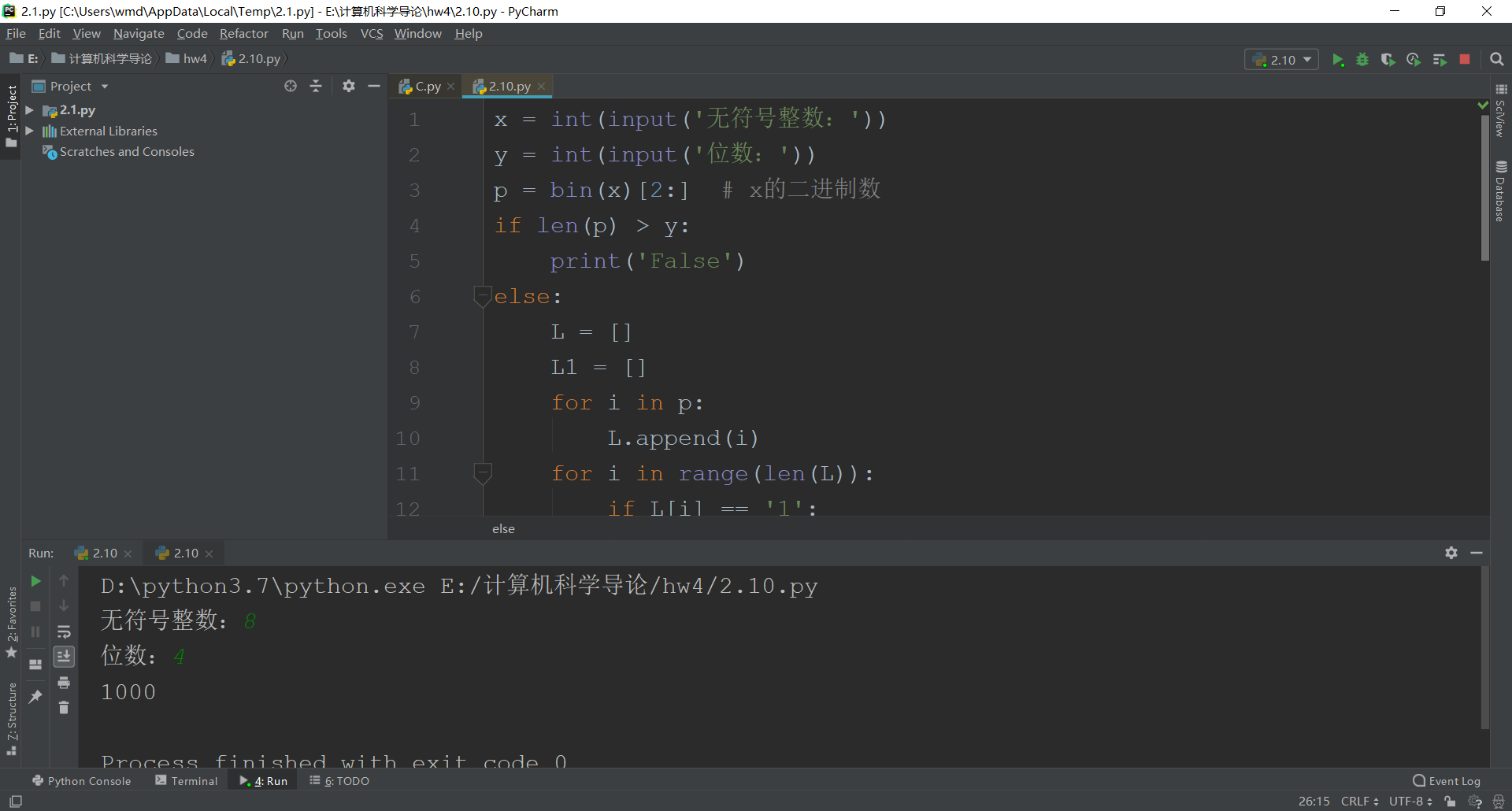
（A）

def fun(p, d=0):  
 if len(p) == 1:  
 if p == '1':  
 d += 1  
 return d  
 else:  
 if p[0] == '1':  
 d += 2\*\*(len(p)-1)  
 p = p[1:]  
 return fun(p, d)  
  
  
b = input('Please enter a binary number:')  
print(fun(b))



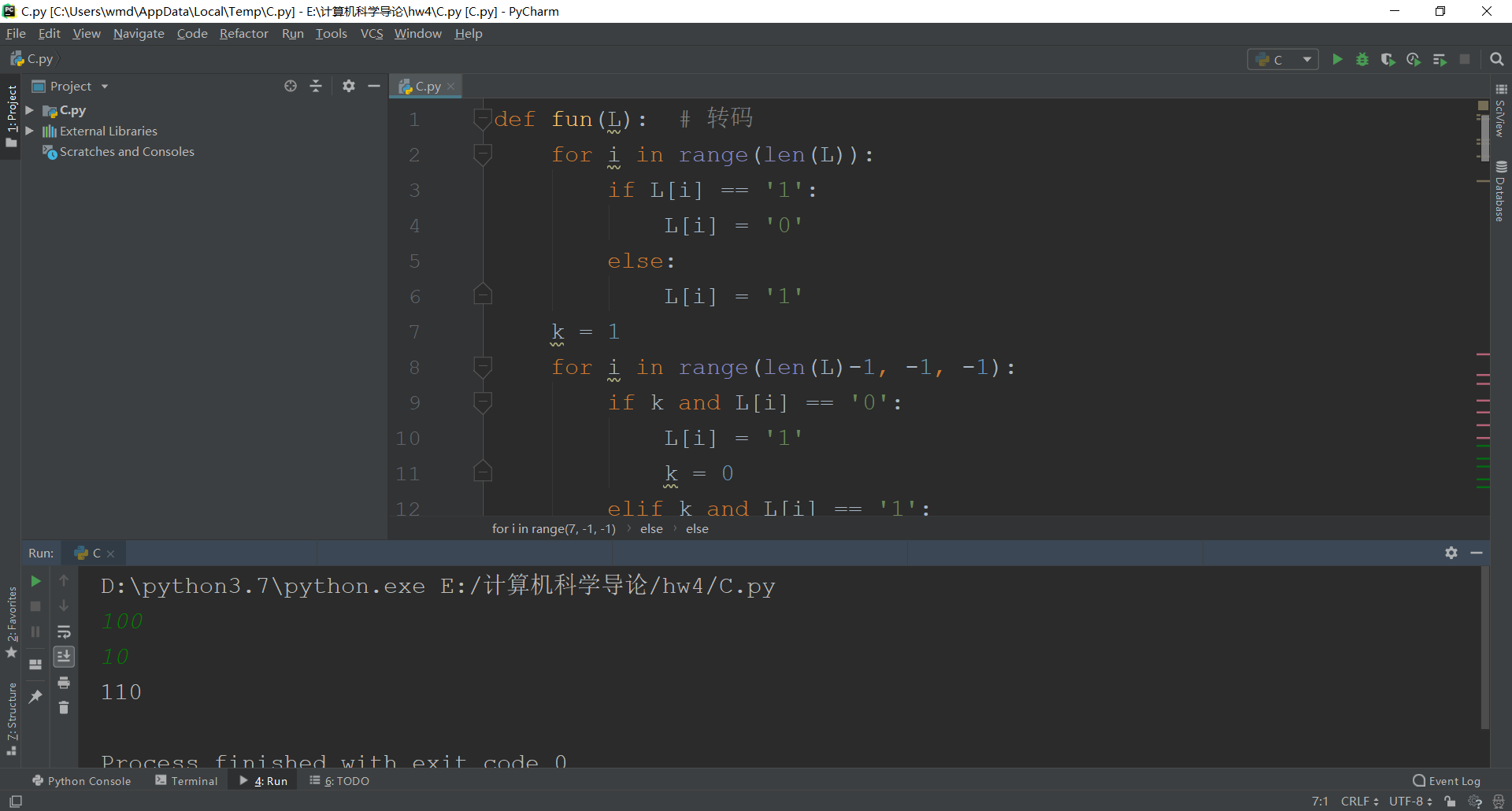
（B）

x = int(input('无符号整数：'))  
y = int(input('位数：'))  
p = bin(x)[2:] # x的二进制数  
if len(p) > y:  
 print('False')  
else:  
 L = []  
 L1 = []  
 for i in p:  
 L.append(i)  
 for i in range(len(L)):  
 if L[i] == '1':  
 L[i] = '0'  
 else:  
 L[i] = '1'  
 k = 1  
 for i in range(len(L)-1, -1, -1):  
 if k and L[i] == '0':  
 L[i] = '1'  
 k = 0  
 elif k and L[i] == '1':  
 L[i] = '0'  
 ans = ''  
 for each in L:  
 ans += each  
 print(ans)



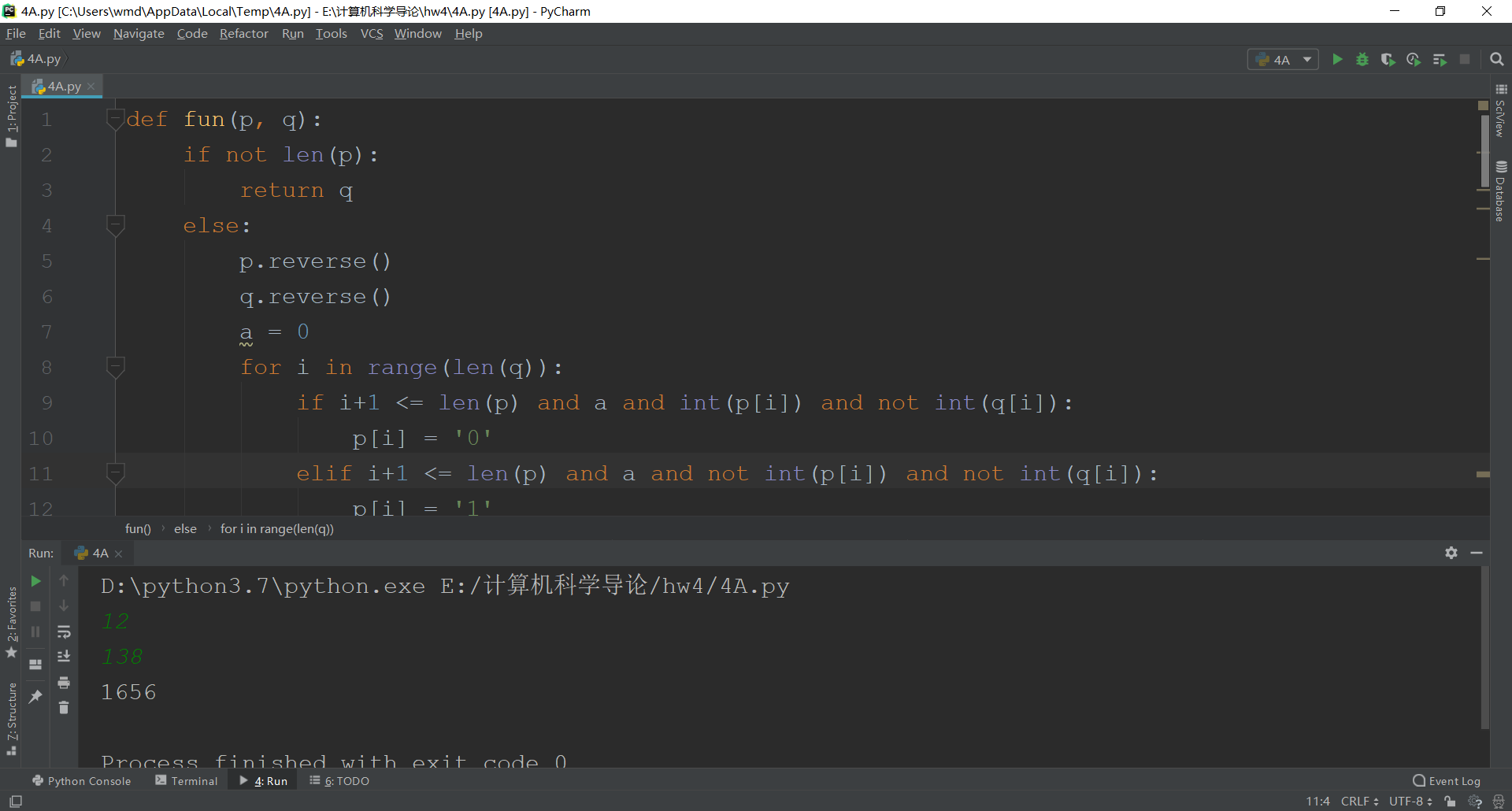
（C）

def fun(L): # 转码  
 for i in range(len(L)):  
 if L[i] == '1':  
 L[i] = '0'  
 else:  
 L[i] = '1'  
 k = 1  
 for i in range(len(L)-1, -1, -1):  
 if k and L[i] == '0':  
 L[i] = '1'  
 k = 0  
 elif k and L[i] == '1':  
 L[i] = '0'  
 return L  
  
  
def fun2(s): # 转十进制  
 r = int(s, 2)  
 if r >= 128:  
 return r-256  
 else:  
 return r  
  
  
x = int(input())  
y = int(input())  
a = 0  
if x < 0:  
 a = 1  
 x = -x  
i = 128  
k = []  
while i >= 1:  
 if x // i:  
 k.append('1')  
 x = x - i  
 else:  
 k.append('0')  
 i //= 2  
if a:  
 k = fun(k) # k为x  
  
b = 0  
if y < 0:  
 b = 1  
 y = -y  
i = 128  
s = []  
while i >= 1:  
 if y // i:  
 s.append('1')  
 y = y - i  
 else:  
 s.append('0')  
 i //= 2  
if b:  
 s = fun(s) # s为y  
  
ans = '' # 开始加法  
a = 0  
for i in range(7, -1, -1):  
 if int(k[i]) and int(s[i]):  
 if a:  
 ans = '1' + ans  
 else:  
 ans = '0' + ans  
 a = 1  
 elif not int(k[i]) and not int(s[i]):  
 if a:  
 ans = '1' + ans  
 a = 0  
 else:  
 ans = '0' + ans  
 else:  
 if a:  
 ans = '0' + ans  
 a = 1  
 else:  
 ans = '1' + ans  
if a and b:  
 if not ans[0]:  
 print('overflow')  
 else:  
 print(fun2(ans))  
elif not a and not b:  
 if ans[0]:  
 print('overflow')  
 else:  
 print(fun2(ans))  
else:  
 print(fun2(ans))



4/A

def fun(p, q):  
 if not len(p):  
 return q  
 else:  
 p.reverse()  
 q.reverse()  
 a = 0  
 for i in range(len(q)):  
 if i+1 <= len(p) and a and int(p[i]) and not int(q[i]):  
 p[i] = '0'  
 elif i+1 <= len(p) and a and not int(p[i]) and not int(q[i]):  
 p[i] = '1'  
 a = 0  
 elif i+1 <= len(p) and not a and int(p[i]) and int(q[i]):  
 p[i] = '0'  
 a = 1  
 elif i+1 <= len(p) and not a and not int(p[i]) and int(q[i]):  
 p[i] = '1'  
 elif i+1 > len(p):  
 if a and int(q[i]):  
 p.append('0')  
 elif a:  
 p.append('1')  
 a = 0  
 elif int(q[i]):  
 p.append('1')  
 else:  
 p.append('0')  
 if a:  
 p.append('1')  
 p.reverse()  
 return p  
  
  
x = bin(int(input()))[2:]  
y = bin(int(input()))[2:]  
L1 = [i for i in x]  
L2 = [i for i in y]  
a = len(L1)  
b = len(L2)  
if a > b:  
 a = L1  
 b = L2  
else:  
 a = L2  
 b = L1 # a是大的，b是小的  
L = []  
k = 1  
for i in range(len(b)):  
 if b[len(b)-i-1] == '1':  
 lll = a + ['0']\*i  
 L = fun(L, lll)  
ans = ''  
for each in L:  
 ans = ans + each  
print(int(ans, 2))



4/B

def fun1(L): # 转码  
 for i in range(len(L)):  
 if L[i] == '1':  
 L[i] = '0'  
 else:  
 L[i] = '1'  
 k = 1  
 for i in range(len(L)-1, -1, -1):  
 if k and L[i] == '0':  
 L[i] = '1'  
 k = 0  
 elif k and L[i] == '1':  
 L[i] = '0'  
 return L  
  
  
def fun2(p, q): # 加法  
 if not len(p):  
 return q  
 else:  
 p.reverse()  
 q.reverse()  
 a = 0  
 for i in range(len(q)):  
 if i+1 <= len(p) and a and int(p[i]) and not int(q[i]):  
 p[i] = '0'  
 elif i+1 <= len(p) and a and not int(p[i]) and not int(q[i]):  
 p[i] = '1'  
 a = 0  
 elif i+1 <= len(p) and not a and int(p[i]) and int(q[i]):  
 p[i] = '0'  
 a = 1  
 elif i+1 <= len(p) and not a and not int(p[i]) and int(q[i]):  
 p[i] = '1'  
 elif i+1 > len(p):  
 if a and int(q[i]):  
 p.append('0')  
 elif a:  
 p.append('1')  
 a = 0  
 elif int(q[i]):  
 p.append('1')  
 else:  
 p.append('0')  
 if a:  
 p.append('1')  
 else:  
 p.append('0')  
 p.reverse()  
 return p  
  
  
x = bin(int(input('被除数：')))[2:]  
y = bin(int(input('除数：')))[2:]  
L1 = []  
L2 = []  
for each in x:  
 L1.append(each)  
for each in y:  
 L2.append(each)  
i = 0  
while 1:  
 f = fun2(fun1(['0']\*(len(L1)-len(L2)+1)+L2), ['0']+L1)  
 g = 1  
 for j in range(1, len(f)):  
 if j == 1:  
 g = 0  
 break  
 if len(L1) < len(L2) or f[1] == '1' or g:  
 break  
 a = len(L1)  
 L1 = f  
 if len(L1) > a:  
 del(L1[0])  
 while not int(L1[0]) and len(L1)-1:  
 del(L1[0])  
 i += 1  
print('商：', end='')  
print(i)  
ans = ''  
for each in L1:  
 ans = ans + each  
print('余数：', end='')  
print(int(ans, 2))

