別紙1

```
# ex1.py on report07-mse
# Haruyuki Ichino
# 2015/06/05
# Calculate inverse element of p'

#coding: UTF-8
import sys
import random

# function
#================================================================================

def calp(p):
    return (2**0)*p[0] + (2**1)*p[1] + (2**2)*p[2] + (2**3)*p[3] + (2**4)*p[4] + (2**5)*p[5] +
(2**6)*p[6] + (2**7)*p[7] + (2**8)*p[8] + (2**9)*p[9] + (2**10)*p[10] + (2**11)*p[11] + (2**12)*p[12]
+ (2**13)*p[13] + (2**14)*p[14] + (2**15)

def is_prime3(q,k=50):
    q = abs(q)
    #judge
    if q == 2: return True
    if q < 2 or q&1 == 0: return False

    #n-1=2^s*d, calc d
    d = (q-1)>>1
    while d&1 == 0:
    ¦   d >>= 1

    #check k times
    for i in range(k):
    ¦   a = random.randint(1,q-1)
    ¦   t = d
    ¦   y = pow(a,t,q)
    ¦   #[0,s-1]の範囲すべてをチェック
    ¦   while t != q-1 and y != 1 and y != q-1:
    ¦   ¦   y = pow(y,2,q)
    ¦   ¦   t <<= 1
    ¦   if y != q-1 and t&1 == 0:
    ¦   ¦   return False
    return True

# Expanded Euclidean
def EuclideanPlus(a, b):
    (xp, xn) = (0, 1)
    (yp, yn) = (1, 0)
    while b != 0:
    ¦   q = a // b
    ¦   (a, b) = (b, a%b)
    ¦   (xp, xn) = (xn-q*xp, xp)
    ¦   (yp, yn) = (yn-q*yp, yp)

    return (xn, yn, a)

```

```
51 # Calc maltiplicative inverse
52 def calcInv(pd, p):
53     (inv, q, gcd_val) = EuclideanPlus(pd, p)
54     return inv % p
55 
#=============================================================================


56 
57 
58 str = sys.argv[1] #a15,a14,a13,...,a0
59 
60 num = [] # init num
61 #set num
62 for i in range(16):
63     num.append(int(str[15-i])) #num=[a0,a1,a2,...,a15]
64 
65 pd = calp(num)
66 p = pd + 1
67 while(not is_prime3(p)):
68   p=p+1
69 
70 print ("p' = ", pd)
71 print("p = ", p)
72 print ("p'^-1 mod p = ", calcInv(pd, p))
```