1. Acknowledge all sources and collaborations. If you do not give an acknowledgement statement, your assignment may not be graded.

2. Consider the usual recursive (divide the exponent by 2) method for exponentiation.

   (i) Trace the on algorithm on 71^{37} mod 101. Show all intermediate steps.

   (ii) Consider ordinary (not modular) exponentiation. Assume that multiplying two k-bit numbers takes k^2 milliseconds. How long does the recursive algorithm take to compute 1023^{1023} Justify carefully.

3. (i) What number is returned by `zzz(3,0)`? By `zzz(3,1)`? By `zzz(3,2)`?

   (ii) For what integers $y < 100$ does `zzz(x,y)` always return x to the power y? Justify briefly.

```
def zzz(x,y): #integers, y >= 0
  if y==0: return 1
  z = zzz(x,y/2)
  return (x*z*z)
```

4. (i) Explain briefly why `isp( )` is correct.

   (ii) Let $k = \lg n$. As a function of k and/or n, give the $\Theta(\,)$ runtime of `isp( )`.

   (iii) Run `probp( )` with $t = 1$ and n all integers at least 2 and less than 1000. Record (a) when the input is prime, how often it makes an error (b) when the input is composite, how often it makes an error.

   (iv) Repeat (iii) with $t = 10$.

```
def isp(n):
  if (0==n%2): return False
  d = 3
  while (d*d < n):
    if (0==n%d): return False
    d += 2
  return True

def probp(n,t):
  for _ in range(t):
    a = random.randint(2,n-1)
    x = pow(a,n-1,n)
    if (1!=x): return False
  return True
```

5. (i) Let d, a, x, b, y be integers. Assume d divides a and b. Prove d divides $ax + by$.

   (ii) Further assume $ax + by > 0$. Prove or disprove: $d \leq ax + by$.

   (iii) Let $a = 16673011647$. Let $b = 16213295811$. Using the fact that $a * -77566962 + b * 79766315 = 51$, prove that 51 is the gcd of a and b.

   (iv) Let $a = 35267$. Let $b = 2161$. Using the fact that $a * 4641 - b * 75740 = 7$, prove or disprove that 7 is the gcd of a and b.