1. (50 points) Use induction to prove the following: The sum of the arithmetic series with the first term $a_1$ and increment $d$ is $a_1 + a_2 + \ldots + a_n = \sum_{i=1}^{n} a_1 + d(i - 1) = \frac{1}{2} n [2a_1 + d(n - 1)]$

A term of the arithmetic series is computed using: $a_k = a_{k-1} + d$. We can compute that $a_k = a_{k-1} + d = a_{k-2} + 2d = \ldots = a_1 + d(k - 1)$. This is where the above formula comes from.

Clearly label the basis, inductive hypothesis, and inductive step.

2. (50 points) Consider the following Python program and use induction on the number of iterations of the for loop to prove that the program always prints the following line.

**True:** y is equal to z.

You must present the proof based on the code and on the number of iterations of the for loop. Clearly label the basis, inductive hypothesis, and inductive step. The loop is an implementation of the above question.

```python
import operator
a1 = input('Pick the starting number: ')
print ('You entered a1 = %s. 
' % a1)
d  = input('Pick the increment: ')
print ('You entered d = %s. 
' % d)
n = input('Pick a number greater than 0: ')
print ('You entered %s. 
' % n)
y = 0
x = a1
for i in range (1,n+1): # executes for i = 1, 2, ..., n
    y = y + x # Sum of the terms so far is in y.
    print ('y is %s. 
' % (y))
    x = x + d # Compute the next term
    print ('x is %s. 
' % (x))

z = (n * (( 2 * a1) + d * (n - 1))) / 2
print ('z is %s. 
' % (z))

if operator.eq (y,z):
    print "True: y is equal to z."
else:
    print "False: y is not equal to z."
```