There are four questions on both sides of this page, with question 1 having 4 parts, question 2 having 6 parts and question 4 having 3 parts. Answer all parts of each question. The points listed for each question indicate the weight they will be given in averaging the grade, but may not be an indication of how long it should take to answer. However, don't spend more than 1 minute per point until you have tried them all. There are 100 points and 80 minutes in the exam. No notes are permitted for the exam. The ANT instruction set is on the front screen for your reference.

1. (18 points) Two of the following four Python functions (found without comments, tsk, tsk) have the same effect. That is, they return the same value when given the same STRING value as an argument. In a sentence, what do they do? The other two either return a different result or cause an error. Which ones are they and what do they do? Suggest a simple change to each of the deviant two that would make its effect the same as the two with identical values returned. You should only modify one command in each, neither add nor delete a command.

a. def fee(x):
   y=""
   for i in x:
      y=i+y
   return y

c. def foo(x):
   y=""
   i=0
   while i<len(x):
      y=x[i]+y
      i=i+1
   return y

b. def fie(x):
   y=""
   for i in range(len(x)):
      y=i+y
   return y

d. def fum(x):
   y=""
   for i in range(len(x)):
      y=y+x[-i]
   return y

2. (12 points) In an 8 bit machine, like ANT, what are the signed binary and hexadecimal equivalents of the following decimal numbers?

a. 2    b. -2    c. 42    d. 127    e. -127    f. 65

3. (20 points) Write an ANT program in ANT assembler language that, when assembled and run, inputs a sequence of characters from the ASCII port and halts when a character that is NOT a lower case letter, a through z, is inputted. You do not need to output each character as they are inputted, but, before halting your program, you should output the offending character to the ASCII port. This is a SHORT program with one loop using only 4 registers and less than 10 commands total. Don’t get carried away. In writing the answer make sure the 6 columns of the assembler line, Name through Comments, are clear.

4a. (20 points) Write a Python function that accepts 2 strings of equal length and returns the number of positions in which they differ. For example, if your function were called, distance, then distance("AGGACATTA","AGTACAGTA") would return 2 since the strings differ in positions [2] and [7], but are equal in all other positions. As an aside, such a function could be used to measure the distance between 2 DNA sequences, although that process also involves lining up similar segments of longer sequences. You are not asked to do that here.

OVER!!!!
4b. (15 points) Use your function from part 4a as a sub-function to write a Python function that accepts a string and a list of strings as arguments, all strings of equal length, and returns the POSITION of the string in the list that is most similar, that is, has the smallest number of character differences between them. If there is a tie, that is, several strings in the list are equally similar, just return the position of any one, not all of them. If the list is empty, your function should return -1.

4c. (15 points) Use your function from part 4a as a sub-function to write a Python function that accepts a string and a list of strings and returns the list with all of the strings in it that differ in more than 5 positions from the given string, if any, removed from the list.