Problem 13 (10 points)
Draw a diagram that illustrates how the bubblesort algorithm would sort this list. Draw the contents of the list after each full pass of the bubblesort algorithm through the list (not each swap). Underline any numbers that are guaranteed to be sorted. (Do NOT use the "Smart" bubblesort algorithm that knows when to stop early.)

[2, -3, 45, 10, -45, 100]

Problem 14 (10 points)
Draw a diagram that illustrates how the mergesort algorithm would sort this list. Draw the contents of the list after each splitting and merging step of the algorithm:

[2, -3, 45, 10, -45, 100, 1000]

Name: _____

Problem 15A (10 points)
Draw a diagram that shows how the binary Search algorithm would behave when looking for the number 162 in the following sorted list. At each step, circle the list element that binary Search compared 162 against, and then show which half of the list remains:

[2, 3, 10, 25, 100, 160, 162, 163, 170, 200, 201]

Problem 15B (4 points)
How many comparisons in total did the binary search need to make before it found 162? _____
How many comparisons would a sequential search have needed to take? _____

8. Computational Complexity: Stock Games (10 points):

You are hired by a BigWinner Inc. to finish their stock recommendation software package after the previous developer was hit by a bus. The previous developer has left you two functions (RateStocksA, and RateStocksB) which are used to predict how much profit stocks will give in the next day. You test out each function with 1 stock, 2 stocks, and 5 stocks and find the following run-time and prediction accuracy results:

Number of Stocks	RateStocksA	RateStocksB
1	5 seconds / 89 % accuracy	1 second / 92% accuracy
2	10 seconds / 90 % accuracy	4 seconds / 92.5 % accuracy
5	25 seconds / 89.5 % accuracy	25 seconds / 91.9 % accuracy

a. What is the Big O (Computational Complexity) class of each function (N = Number of Stocks) (6 points)?

RateStocksA _____ ; RateStocksB _____

b. Assuming you want to analyze the top 2500 stocks on the NYSE at the end of one trading day and decide what to purchase by the start of the next day to maximize your profit, which algorithm would you use? WHY? (4 points)

9. Know your sequences! (9 points)

Three of the compound data types you have learned about are sequences. Name these three different types of sequences, give an example of each, and **state why they are different** from each other.

1.

9. *(14 points)*

You are going to have a dinner party where you invite N guests. If N=5, the guests will be numbered [0,1,2,3,4]. None of the guests know each other, so you write the following code to "introduce" each guest to every other guest.

```
def introduce(GuestA, GuestB):
   print GuestA, " I'd like to introduce you to", Guest B
   print GuestB, "meet", GuestA

def dinnerParty( listOfGuests ):
  for guestX in listOfGuests:
    for guestY in listOfGuests:
       introduce( guestX, guestY)

dinnerParty( [0,1,2,3,4] )
```

Notice that the above code introduces the same guest to themselves, and also introduces a pair of guests twice (it introduces 0 to 1, and then 1 to 0). This is not exactly the same as a dinner party with real humans.

Your question: If you assume that a call to the introduce(...) function is your unit of work (i.e. just like a comparison in a sorting algorithm), what is the Big O complexity class of this problem? In other words, as the number of guests (N) increases, how quickly does the number of introductions increase?

```
Fill in the following blanks (2pts each):

if N = 2   the number of Introductions =   ________

if N = 4   the number of Introductions =   ________

if N = 8   the number of Introductions =   ________

So therefore, the complexity class is (4pts):   O( _________ )

Also, if it takes 1 second to introduce each pair of guests, how many
seconds will you spend doing introductions if you have 100 guests?

(4pts) ______________________ Seconds
```

Problem 16 (20 points)
Big O() notation is used to describe the computational complexity of algorithms. By looking at how much work an algorithm must perform based upon the number of data items it is working on, Big O() notation can predict which algorithms are the fastest, and which will be slower as the number of data elements (N) increases. In CS 1301 we learned about four general complexity classes.

1. Draw a graph showing the relative curves of the four different complexity classes. The Y axis is the amount of work performed (e.g. number of comparisons) while the X axis represents the number of data elements (N).
2. Label each line on your graph with the appropriate BigO() complexity class.
3. Give an example algorithm that is in that complexity task.

Problem 17 (8 points)
What is 132 in:

Binary (Base 2) : _____

Octal (Base 8): _____

Hexadecimal (Base 16): _____

Decimal (Base 10): _____