## H0428

Name

1. Consider the following incomplete class that stores information about a customer, which includes a name and unique ID (a positive integer). To facilitate sorting, customers are ordered alphabetically by name. If two or more customers have the same name, they are further ordered by ID number. A particular customer is "greater than" another customer if that particular customer appears later in the ordering than the other customer.

```
public class Customer
  // constructs a Customer with given name and ID number
  public Customer (String name, int idNum)
    /* implementation not shown */
  // returns the customer's name
  public String getName()
    /* implementation not shown */ }
  // returns the customer's id
  public int getID()
    /* implementation not shown */ }
  // returns 0 when this customer is equal to other;
  // a positive integer when this customer is greater than other;
       a negative integer when this customer is less than other
  11
 public int compareCustomer(Customer other)
  { /* to be implemented in part (a)
  // There may be fields, constructors, and methods that are not shown.
)
```

a. Write the Customer method compareCustomer, which compares this customer to a given customer, other. Customers are ordered alphabetically by name, using the compareTo method of the String class. If the names of the two customers are the same, then the customers are ordered by ID number. Method compareCustomer should return a positive integer if this customer is greater than other, a negative integer if this customer is less than other, and 0 if they are the same.

For example, suppose we have the following Customer objects.

• Customer c1 = new Customer("Smith", 1001);

- Customer c2 = new Customer("Anderson", 1002);
- Customer c3 = new Customer("Smith", 1003);

The following table shows the result of several calls to compareCustomer.



Method Call	Result
c1.compareCustomer(c1)	0
c1.compareCustomer(c2)	a positive integer
c1.compareCustomer(c3)	a negative integer

Complete method compareCustomer below.

// returns 0 when this customer is equal to other; // a positive integer when this customer is greater than other; // a negative integer when this customer is less than other public int compareCustomer(Customer other)

b. A company maintains customer lists where each list is a sorted array of customers stored in ascending order by customer. A customer may appear in more than one list, but will not appear more than once in the same list.

Write method prefixMerge, which takes three array parameters. The first two arrays, list1 and list2, represent existing customer lists. It is possible that some customers are in both arrays. The third array, result, has been instantiated to a length that is no longer than either of the other two arrays and initially contains null values. Method prefixMerge uses an algorithm similar to the merge step of a Mergesort to fill the array result. Customers are copied into result from the beginning of list1 and list2, merging them in ascending order until all positions of result have been filled. Customers who appear in both list1 and list2 will appear at most once in result.

list1	Arthur	Burton	Burton	Franz	Horton	Jones	Miller	Nguyen
	4920	3911	4944	1692	9221	5554	9360	4339
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
								-
list2	Aaron	Baker	Burton	Dillard	Jones	Miller	Noble	
	1729	2921	3911	6552	5554	9360	3335	
	[0]	[1]	[2]	[3]	[4]	[5]	[6]	-
result	null	null	null	null	null	null		
	[0]	[1]	[2]	[3]	[4]	[5]		

For example, assume that three arrays have been initialized as shown below.

In this example, the array result must contain the following values after the call



## H0428

prefixMerge(list1, list2, result).

result	Aaron	Arthur	Baker	Burton	Burton	Dillard
	1729	4920	2921	3911	4944	6552
	[0]	[1]	[2]	[3]	[4]	[5]

In writing prefixMerge, you may assume that compareCustomer works as specified, regardless of what you wrote in part (a). Solutions that create any additional data structures holding multiple objects (e.g., arrays, ArrayLists, etc.) will not receive full credit.

Please respond on separate paper, following directions from your teacher.