Name _____ Period_____

Consider a class that can be used to model a DixieCup, or rather the contents that a DixieCup can hold. When a DixieCup is created the number of items it can hold is defined depending on which constructor is called.

The number of items, if specified, is used to initialize the instance variable itemsArray[] which serves to store the items in the DixieCup as String data types. If a DixieCup can hold items, the instance variable isFull is false when the cup is created, otherwise isFull is true. Consider the following calls,

DixieCup cup1 = new DixieCup(5);//Creates a cup that can hold five items, isFull is false
DixieCup cup2 = new DixieCup(); //Creates a cup that cannot hold anything, isFull is true

(a)   Write the DixieCup class that meets the above requirements below.

```java
public class DixieCup{
    private boolean isFull;
    private String itemsArray[];

    public DixieCup(int t){
        itemsArray = new String[t];
        isFull = false;
    }

    public DixieCup(){
        isFull = true;
        itemsArray = new String[0];
    }
}
```

A *DixieCup* is full if there are no more null values in the *itemsArray*. If there are null values, *isFull* should be set to false, otherwise *isFull* should be set to true. Because we also have DixieCups that cannot hold anything, you will also need to check for this too.

(b)   Write the *setIsFull* method below

```java
    public void setIsFull(){
            isFull = true;
            for(int i = 0; i < itemsArray.length; i++){
                if(itemsArray[i] == null){
                    isFull = false;
                    return;
                }
            }
    }
```

The *addItem* method accepts a parameter which represents the item to be added to the *itemsArray*.   The item you add must go in the first *null* value of your *itemsArray*. You will need a loop to check for this. Once you have added your item, you should call *setIsFull*. *setIsFull* will set *isFull* to *true* if there are not anymore *null* places.

(c)   Write the addItem method

```java
public void addItem(String n){
        for(int i = 0; i < itemsArray.length;i++){

                if(itemsArray[i] == null){
                        itemsArray[i] = n;
                        setIsFull();
                        return;
                }
        }
}
```

In addition to adding items, the user should have the ability to swap the location of two items in a cup. The swap method should have the following signature,

```java
public void swap(int i1, int i2)
```

The parameters, *i1* and *i2*, represent the indices of the items you want to swap.

(d) Write the swap method

```java
public void swap(int i1, int i2){
    String temp = itemsArray[i1];
    itemsArray[i1] = itemsArray[i2];
    itemsArray[i2] = temp;
}
```