Name _____ Period _____

1. This question involves the implementation of a class, called `DixieCup`, which models a dixie cup. A `DixieCup` object can be created with or without a parameter. A `DixieCup` object created with a parameter can hold the number of items specified by the parameter. A `DixieCup` object created without a parameter cannot hold any items.

   - `addItem`, adds an item to a cup that is not full
   - `getIsFull`, returns whether or not the cup is full
   - `getNumItems`, returns the number of items in a cup
   - `swapItems`, swaps the items at the specified locations

Statements and Expressions	**Value Returned (blank if no value)**	**Comment**
`DixieCup cup1 = new DixieCup();`		Creates a cup that cannot hold any items
`DixieCup cup2 = new DixieCup(5);`		Creates a cup that can hold up to 5 items
`cup1.addItem("marble");`		Nothing is added to cup1
`cup2.addItem("marble");`		A marble is added to cup2 and the contents are defined as follows, `{marble, null, null, null, null}`
`cup2.getNumItems();`	1	There is 1 item in cup2
`cup1.getIsFull();`	true	Returns true because cup1 cannot hold any items
`cup2.getIsFull();`	false	Returns false because cup2 can hold more items
`cup2.addItem("marshmallow");`		A marshmallow is added to cup2 and the contents are defined as follows, `{marble, marshmallow, null, null, null}`
`cup1.getNumItems();`	0	There are not any items in cup1
`cup2.getNumItems();`	2	There 2 items in cup2
`cup1.swapItems(0, 1);`		Nothing is swapped in cup1
`System.out.println(cup2.toString())`	marble null marshmallow null null the cup is not full	Returns a summary of the cup

_____/12

Write the complete `DixieCup` class, including the constructors and any required instance variables and methods. Your implementation must meet all specifications and conform to the example.

```
public class DixieCup{

    private boolean isFull;
    private String itemsArray[];

    public DixieCup(){
        isFull = true;
    }

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

    public void addItem(String item){
        if(!isFull){
            for(int i = 0; i < itemsArray.length; i++){
                if(itemsArray[i] == null){
                    itemsArray[i] = item;
                    return;
                }
            }
        }
    }

    public boolean getIsFull(){

        if(itemsArray!=null){
            for(int i = 0; i < itemsArray.length; i++){
                if(itemsArray[i] == null){
                    return false;
                }
            }
        }
        return true;
    }
```

Score _____/12

```
    public int getNumItems(){
        int count = 0;
        if(itemsArray!=null){
            for(int i = 0; i < itemsArray.length; i++){
                if(itemsArray[i] != null){
                    count++;
                }
            }
            return count;
        }
        return 0;
    }

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

    public String toString(){
        String result = "";
        for(int i = 0; i < itemsArray.length; i++){
            result += itemsArray[i] + " ";
            if(getIsFull()){
                result += "The cup is full";
            }else{
                result += "The cup is not full";
            }
        }
        return result;
    }
}
```

/12

Score _____/12