Question - 1

What's the result of test class?

SCORE: **5 points**

```
public class Balloon {

        private String color;

        public Balloon(){}

        public Balloon(String c){
                this.color=c;
        }

        public String getColor() {
                return color;
        }

        public void setColor(String color) {
                this.color = color;
        }
}
public class Test {

        public static void main(String[] args) {

                Balloon red = new Balloon("Red");
                Balloon blue = new
Balloon("Blue");

                swap(red, blue);
                System.out.println("red
color="+red.getColor());
                System.out.println("blue
color="+blue.getColor());

                foo(blue);
                System.out.println("blue
color="+blue.getColor());

        }

        private static void foo(Balloon balloon)
{
                balloon.setColor("Red");
                balloon = new Balloon("Green");
                balloon.setColor("Blue");
        }

        //Generic swap method
        public static void swap(Object o1, Object
o2){
                Object temp = o1;
                o1=o2;
                o2=temp;
        }
}
```

Help

● red color = Red blue color = Blue blue color = Red

○ red color = Blue blue color = Red blue color = Blue

○ red color = Red blue color = Blue blue color = Blue

○ red color = Blue blue color = Red blue color = Green

Question - 2

Quick Union

SCORE: **5 points**

If we are going to apply the quick-find algorithm to solve the dynamic connectivity problem until all components are connected, how many array accesses are necessary?

● N^2

○ NlogN

○ logN

○ N

Question - 3

Stability of insertion Sort

SCORE: **5 points**

In which category does the Insertion Sort algorithm belong to?

● Stable Sort

○ Unstable Sort

Question - 4

Statement

SCORE: **10 points**

The following code will successfully output the Address1@4554617c:

```
// Output: "Address1@4554617c"
public class Address1 {
    public static void main(String[] args) {
        Address1 a1 = new Address1();
        System.out.println(a1);
    }
}
```

When trying to output some additional information, the *toString()* method should be overridden. However, the following code generated a *StackOverflowError* unexpectedly.

```
// Output: "Exception in thread "main"
java.lang.StackOverflowError"
public class Address2 {
    @Override
    public String toString() {
        return "The address of this Object is: "
+ this;
    }

    public static void main(String[] args) {
        Address2 a2 = new Address2();
        System.out.println(a2);
    }
}
```

Please answer the following questions:
(1) (3pt) What caused this error?
(2) (7pt) In order to generate output in the format of "The address of this Object is: Address2@4554617c", what change should be made? (Hint: only a small change is needed).

Note: if you don't know the appropriate Java way to do part 2, just describe in words what you need.

Question - 5

SCORE: **25 points**

Implementation

There is a deck of N cards on the desk and you are going to play a game: take out the card on the top and record it, then put the next card to the bottom of the deck. repeat this process until the deck is empty. The result is the record data.
Please implement a queue to simulate this process, the input will contain the maxsize of the queue and the order of the deck, the size of the deck is no larger than the maxsize of the queue. you should return an array which contains the record of this game.
Sample:
inut: 10, {1,2,3,4,5}
Process: {1,2,3,4,5} -> {3,4,5,2} -> {5,2,4} -> {4,2} -> {2}   Queue
{1} -> {1,3} -> {1,3,5} -> {1,3,5,4} -> {1,3,5,4,2}   ResultArray
output: {1,3,5,4,2}