Question 1 ( 5 pts.)

Suppose s is any String. Write a loop that prints the characters in s backwards, in a column. Thus if s is the string "Who?" your code should print

?  
 o  
 h  
 W

for(int j = s.length()-1; j >= 0; j--)
System.out.println(s.charAt(j));

Question 2 ( 5 pts.)

Write a for loop that prints in a column all numbers between 1 and 200 that are multiples of 3 or multiples of 7.

Your list should start: 3-6-7-9-12-14-15-18-21-24...

Here, 3-6-9-12-15 etc. are the multiples of 3, and 7-14-21 etc. are multiples of 7. Important: NO number in your printed list should appear more than once.

for(int j = 1; j <= 200; j++)
If ((j %3 ==0) || (j %7 == 0)) System.out.println(j);

Question 3 ( 5 pts.)

Suppose s is any String. Write a code fragment, including a loop, that counts the number of r's (upper or lower case) in s, and then prints out that count.

For example, if s is the String "Roar", your code should print 2.

Int ct = 0;
Question 4 (5 pts.)

Suppose str is any String. Write a for loop that prints out the characters of str, but with each character followed by the word "goo".

For example if str is the String "cat", your code should print:

Cgooagogootgoo

for(int j = 0; j<s.length(); j++)
System.out.print(s.charAt(j) + "goo");

Question 5 (5 pts.)

Write a for loop that begins at 0 and prints every other number in a column, down to -100. Thus the first two numbers printed should be 0
-2

and the last two printed should be

-98
-100

for(int j = 0; j >= -100; j = j - 2)System.out.println(j);

Question 6 (25 pts.)

Write a complete program in a single class called RandomAverage, which reads in an int value from the keyboard and store it in variable n. Your program should assume that this value is positive.
Then your program should generate n random numbers using the `random()` method from the `Math` class and print the average of these values. You do not have to print out the random values themselves, and you do not have to format your final output.

Here is some sample behavior:

Enter a positive integer
3
[3 random values are generated, say .12, .21, .66]
[your program then prints]

The average is: .33

```java
import java.util.*;

public class RandomAverage{
  public static void main(String[] args){
    Scanner input = new Scanner(System.in);
    System.out.println("enter size of test:");
    int n = input.nextInt();
    double sum = 0.0;
    for(int k = 0; k < n; k++){
      sum = sum + Math.random();
    }
    System.out.println("The average is: " + (sum/n));
  }
}
```

**Question 7 (25 pts.)**

The EggSellDriver code below makes use of a class called EggSeller; the EggSeller class is intended to model a part of a small egg seller's business.

In the box provided, create a definition for EggSeller that will allow the EggSellDriver code to function as intended.
Here is the output from an EggSellDriver run:

value: 17420.0
value: 18330.0

That is, the value of a batch of 1300 eggs at 13.4 cents / egg, owned by Dana, is $174.20 (17420.0 cents), and at 14.1 cents / egg the value is $183.30.

(Note: Use the "Check Answer" button beneath the box to make your submission, which you may revise repeatedly while the exam is active. That button will NOT provide any information about the quality or correctness of your submission.)

```java
public class EggSellDriver{
    public static void main(String[] args){
        EggSeller dana = new EggSeller("Dana",1300,13.4);
        System.out.println("value: "+ dana.getPrice()*dana.getEggCount());
        dana.setPrice(14.1);
        System.out.println("value: "+ dana.getPrice()*dana.getEggCount());
    }
}

public class EggSeller
{
    private String seller;
    private int eggCount; // starting Bid at auction
    private double price; // what painting sold for

    public EggSeller(String seller,int c, double p)
    {
        this.seller = seller;
        eggCount = c;
        price = p;
    }

    public String getSeller(){return seller;}
```
public int getEggCount() {return eggCount;}
public double getPrice() {return price;}
public void setPrice(double amt){price = amt;}

Question 8 (5 pts.)

A goblet is a fancy, old-fashioned name for a high-end drinking glass. Suppose you are in the goblet business, and your business requires some software support in Java, so you've created a Goblet class, given below. Notice that a Goblet object consists of: a set or batch of goblets of a certain count; a size for each goblet in the batch; and a certain price for each goblet in the batch.

The next five questions are based on the Goblet class.

This class as written has an incomplete constructor. In the box provided, enter a complete constructor definition for the Goblet class.

public class Goblet{
    private int size; // size, in ounces
    private double price; // price of a single goblet in a batch
    private int count; // number of goblets in a batch

    public Goblet(int s, double p, int count){
        // your constructor body goes here
    }

    public int getSize(){return size;}
    public double getPrice(){return price;}
    public int getCount(){return count;}

    public void setPrice(double newPrice){price = newPrice;}

}

public Goblet(int s, double p, int count){
    size = s; price = p; this.count = count;}

Question 9 (5 pts.)

Now suppose you are writing a separate driver class in which to test your Goblet code. Write a single statement that would go inside `main` in the driver, and that would create a Goblet object called `todaysGoblets`. It should consist of 100 goblets; each should hold 6 ounces of liquid; and each individual goblet should cost $11.95.

```java
Goblet todaysGoblets = new Goblet(6, 11.95, 100);
```

Question 10 (5 pts.)

Now suppose again that you are implementing a separate driver class in which to test your Goblet code. Suppose that in the driver there is a Goblet object called `someGoblets`.

Now write one or several statements in the box below that will be in the driver, and that raise the price of each goblet in the `someGoblets` object by 10%.

```java
someGoblets.setPrice(1.1 * someGoblets.getPrice());
```

Question 11 (5 pts.)

Now suppose you decide to extend the set of methods in the Goblet class by adding a `setCount` method. This method should allow you to change the number or count of goblets in a Goblet object. In the box provided below, write a definition for this method that will go inside the Goblet class.

```java
public void setCount(int c) { count = c; }
```

Question 12 (5 pts.)

Now suppose you decide to extend the set of methods in the Goblet class by adding a method with no arguments, called `batchPrice`,
which returns the price of the full batch. Thus if the price of a single goblet in a batch is $4.20, and if the batch contains 100 goblets, then batchPrice in this case would return the product of 4.20 and 100, or 420.00.

    public double batchPrice(){return count*price;};