Rutgers Honor Pledge

Rutgers honor pledge: *On my honor, I have neither received nor given any unauthorized assistance during this examination.*

You are on your honor to complete this quiz without assistance. You MAY NOT ask any human for assistance or use any online tutoring service.

By continuing, you are AGREEING TO THE ABOVE HONOR STATEMENT and Rutgers Academic Integrity policies.

Problem 1: Arrays & Functions (25 points)

What will be printed as a result of executing the following Java program? (5 points)

```java
public class ProgramFinal{
    Run | Debug
    public static void main(String[] args) {
        int[] b = {11, 20, 7, 12, 15, 69, 42, 21, 39};
        mystery2(3, b);
        printArray(b);
    }
    public static void printArray(int[] b){
        for(int elt : b){
            System.out.print(elt + " ");
        }
        System.out.println();
    }
    public static void mystery2(int x,  int[] nums){
        for (int k = 1; k < nums.length; k = k + x){
            nums[k] = nums[k - 1] + x;
        }
    }
}
```

Answer: 11 **14** 7 12 **15** 69 42 **45** 39 - (2 pts for non-bolded items - 3 pts for bolded items)

(20 points) A *FooGame Tournament* is a single elimination tournament. Once a player loses, the player is out. Each round has half as many players as the round before. The rounds continue until there is ONE winner.

In our simulated tournament:
- a `String` array is filled with player names. An `int` array is filled with player scores such that names[0] has their score recorded in scores[0], names[1] has his/her score recorded in scores[1], etc.
- ~~There are 16 players in the tournament.~~ The number of players is a power of 2 input by the user.
- The player scores are randomly generated such that the players in the even positions have even scores (0 is considered even), and the players in the odd positions have odd scores.
- Player 0 plays against Player 1, Player 2 plays against Player 3, etc.
- The player with the higher of the two scores advances to the next round. In the diagram above, Kevin had a higher score than Adrian and therefore advanced to the next round. The incomplete program code is below.
- The game is mostly written for you. You are tasked with writing **THREE** methods:
  - **(5 pts) generateEven**: has no parameters; generates and returns a random **even** integer in the range 0 – 98 inclusive.
  - **(5 pts) generateOdd**: has no parameters; generates and returns a random **odd** integer in the range 1 – 99 inclusive.
  - **(10 pts) nextRound**: This method has two parameters: a String array, *players*, containing the names of players and an int array, *theScores*, containing the scores of the players such that players[0] has score theScores[0], players[1] has score theScores[1], etc. This method returns a

String array containing the winners of the tournament games for that round. The array that is returned is half the size of the parameter array. The tournament pairings are consecutive array elements (i.e. players[0] and players[1] play against each other; players[2] and players[3] play against each other, etc.

The program code is below.

```java
public class Game{
    public static void main(String[] args){
        int numPlayers = Integer.parseInt(args[0]); // This number is a power of 2 input by the user.
        String[] names = new String[numPlayers];
        for(int k = 0; k < names.length; k++){
            names[k] = StdIn.readString(); // Player names
        }

        while (names.length > 1){
            System.out.println("\nCurrent round of players:";
            int[] scores = new int[names.length];
            for (int i = 0; i < names.length; i++){
                if (i%2 == 0){
                    scores[i] = generateEven();
                }else{
                    scores[i] = generateOdd();
                }
            }
            for (int j = 0; j < names.length; j++){
                System.out.println(names[j] + "\t\t" + scores[j]);
            }
            names = nextRound(scores,names);
        }

        System.out.println(names[0] + " is the winner!");
    }
```

The game: Each iteration of the while loop will generate the scores of the players in a round of the tournament. Each round has half as many players. You are to write the code for the three methods:
- generateEven
- generateOdd
- nextRound

```java
// Generates and returns an even integer in the range of 0 to 98 inclusive.
public static int generateEven(){
    //You need to write this code
}
// Generates and returns an odd integer in the range of 1 to 99 inclusive.
public static int generateOdd(){
    //You need to write this code
```

```
    }

/* This method has two parameters: a String array, **players**, containing the names of players and an int array, **theScores**, containing the scores
of the players such that players[0] has score theScores[0], players[1] has score theScores[1], etc.
This method returns a String array containing the winners of the tournament games for that round. The array that is returned is half the size of
the parameter array. The tournament pairings are consecutive array elements (i.e. players[0] and players[1] play against each other; players[2]
and players[3] play against each other, etc.
*/
  public static String[] nextRound(int[] theScores, String[] players){
      //You need to write this code
  }
}
```

The command **java Game 16 < players.txt** may result with the following display:

```
Current round of players:
Adrian          12
Kevin           57
Lisa                        36
Scott           11
Bert                        64
Arthur          67
Drew            38
Ernie           31
Mike            42
Andrew          99
Johan           38
Ian                         73
Monica          62
Steve           15
Red             54
Robert          9

Current round of players:
Kevin           66
Lisa                        43
Arthur          22
Drew            87
Andrew          32
Ian                         91
Monica          70
Red             1

Current round of players:
Kevin           62
Drew            21
Ian                         12
Monica          53


Current round of players:
Kevin           30
Monica          83

Monica is the winner!
```

Solution:

```java
public static int generateEven(){
    int randEven = (int)(Math.random() * 50) * 2;
    return randEven;
 }
  public static int generateOdd(){
    int randOdd = ((int)(Math.random() * 50) * 2) + 1;
    return randOdd;
}
  public static String[] nextRound(int[] a, String[] n){
    String[] s = new String[a.length/2];
    int count = 0;
    for(int i = 0; i < a.length; i += 2){
        if (a[i] > a[i+1]){
            s[count] = n[i];
        }
        else{
            s[count] = n[i + 1];
        }
        count++;
    }
    return s;
}
```

(5 pts) generateEven
1 pt for returning a generated random int.
3 pts for generating an even int
1 pt for the correct range of even ints.

(5 pts) generateOdd
1 pt for returning a generated random int.
3 pts for generating an odd int
1 pt for the correct range of odd ints.

(10 pts) nextRound
2 pts for creating new array that is half the size
3 pts for comparing consecutive terms of parameter array-check loop-could be either array length
3 pts for correct assignment to the new array based on array a.
2 pt for return the created array.

Rutgers Honor Pledge

Rutgers honor pledge: *On my honor, I have neither received nor given any unauthorized assistance during this examination.*

You are on your honor to complete this quiz without assistance. You MAY NOT ask any human for assistance or use any online tutoring service.

By continuing, you are AGREEING TO THE ABOVE HONOR STATEMENT and Rutgers Academic Integrity policies.

Problem 2: Recursion (25 points)

1. (12 pts) Consider the recursive function foo GIVEN IN PSEUDOCODE below:

```
foo(int x, int y) {
    if (y is 0)   return 1;
    if (y is even) return foo(x*x, y/2);
    else return foo(x*x, y/2) * x;
}
```

a. (3 pts) What is the output if foo(2, 0) is called?
   a. ANSWER.   1
b. (3 pts) What is the output if foo(3,2) is called?
   a. ANSWER.   9
c. (3 pts) What is the output if foo(2,3) is called?
   a. ANSWER.   8
d. (3 pts) Explain what foo does? That is, what does it compute given x and y?
   a. The code computes the power function foo(x,y) = x^y

2. (18 pts) Define a H-tree of order n as a combination of one H-tree of order n and 4 H-trees of order (n-1) etc connected to each tip of H-tree of order n. Each 'H' in a H-tree is drawn by connecting 3 lines of equal length. The H-trees of order (n-1) are half the size of H-tree of order n. As an example, here are the H-trees of orders 1, 2, 3, 4, and 5. H-tree of order 0 is null.
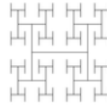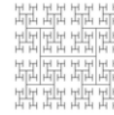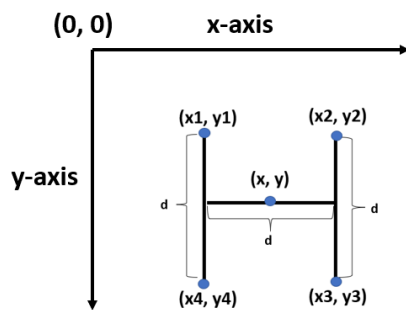

Order 1   order 2   order 3   order 4   order5

a. (6 pts – 1.5 pts each) An H-tree of order 1, centered at (x,y) is shown below. The length of each side of this tree is d. Find the coordinates of the 4 tips of the 'H' using (x, y) and d. That is, compute the coordinates (x1, y1), (x2, y2), (x3, y3) and (x4, y4) using (x, y) and d. Note that the origin (0, 0) of the coordinate system is at upper-left corner.

$(x1, y1) =$ _____(x - d/2, y - d/2)_____

$(x2, y2) =$ _____(x + d/2, y - d/2)_____

$(x3, y3) =$ ___(x + d/2, y + d/2)_____

$(x4, y4) =$ ___(x – d/2, y + d/2)_____

b. (12 pts) Suppose you have implemented a recursive H-tree drawing program using the code

```
void draw(int n, double x, double y, double d) {
        if (n == 0) return;
      /* code to draw 3 lines to make 'H' that centers at (x, y) with
        Each line of length equals to d.
      */
        void drawH(double x, double y, double d) {
          /* code to draw a H-tree of order n */


          /* first recursive call to draw a H of order (n-1)*/

          /* second recursive call to draw a H of order (n-1)*/

          /* third recursive call to draw a H of order (n-1)*/

          /* fourth recursive call to draw a H of order (n-1)*/
}
```
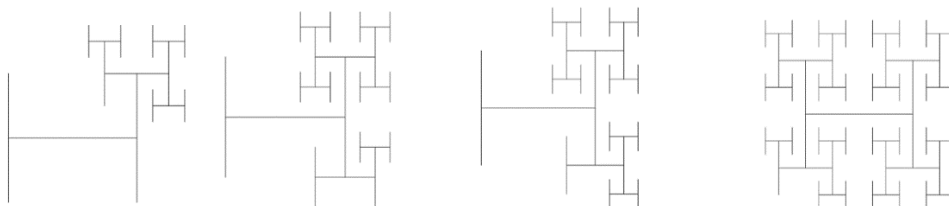
   Although above recursive calls can be done in any order, we would like to do them in an order that produces the trees as shown below for n=3 (H-tree of order 3). The images below display the <u>interim stages</u> of the H-tree of order-3 creation. Each one should provide a clue as to what order, the 4 trees of order (n-1) are drawn using each tip as the center.

What is the order of 4 recursive calls to draw with proper arguments? (3 pts each - if order of calls and arguments are correct. If arguments are correct, but order of calls wrong, max credit – 6 pts)

(i)   draw(n-1 , x + d/2 , y - d/2 , d/2);
(ii)   draw(n-1 , x + d/2 , y + d/2 , d/2);
(iii)   draw(n-1 , x - d/2 , y - d/2 , d/2);
(iv)   draw(n-1 , x - d/2 , y + d/2 , d/2);

image deleted

Problem 3: OOP (40 points)

3.1 (16 points)
Given the following method signatures and instance variables, complete the implementation of the following RaceCar class. A RaceCae object can increase its speed, decrease its speed, increase or decrease its fuel, empty its fuel, and can check the current speed or fuel. The breakdown will be as follows:

- Implement the RaceCar Constructor method with your own parameters: 4 points
- Increase speed: 2 points
- Decrease speed: 2 points
- Increase Fuel: 2 points
- Decrease Fuel: 2 points
- Empty Fuel: 2 points
- Get Speed: 1 point
- Get Fuel: 1 points

Begin by Initializing the race car such that the current speed is 0.

```java
public class RaceCar{

    private int currentFuel; //in gallons
    private int currentSpeed; //in mph
    private int speedLimit; // the max speed limit the car can go
    private int maxFuelCapacity; // the maximum number of gallons the
car tank can hold

    public RaceCar(){
    //implement this method'
        this.currentFuel=currentFuel;
        this.currentSpeed=currentSpeed;
        this.speedLimit=speedLimit;
            this.maxFuelCapacity=maxFuelCapacity;
    }
    public boolean increaseSpeed(int speedFactor){
            //implement this method that will increase the speed and
    return true if speed was successfully increased, false otherwise
            if(currentSpeed+speedFactor<=speedLimit){
                currentSpeed+=speedFactor;
```

```java
                return true;
            }
        return false;
    }
    public boolean decrease(int speedFactor){
                //implement this method. Return true if speed was
    successfully decreased, false otherwise
            if(currentSpeed-speedFactor>=0){
                currentSpeed-=speedFactor;
                 return true;
            }
        return false;
    }
    public boolean increaseFuel(int fuel){
        //implement this method. Return true if fuel was
    successfully added, false otherwise
            if(currentFuel+fuel<=maxFuelCapacity){
                currentFuel+=fuel;
                    return true;
            }
        return false;
    }
    public boolean decreaseFuel(int fuel){
            //implement this method. Return true if fuel was
    successfully decreased, false otherwise
            if(currentFuel-fuel>=0){
                currentFuel-=fuel;
              return true;
            }
          return false;
    }
    public void emptyFuel(){
        //implement this method to deplete the tank of all fuel
        this.currentFuel=0;
    }
    public int getSpeed(){
        //return the current speed of the race car
        return this.currentSpeed;
    }
    public int getFuel(){
        //return the current amount of fuel in the race car
        return this.currentFuel;
    }
}
```

3.2

```
public class BandMember
{
    private String name;
    private int age;

    public BandMember()
    {  /*  Implementation not shown */  }

    public BandMember(String name, int age)
    {  /*  Implementation not shown */  }

    // no other constructors
}
```

Which of the following declarations will compile without error? (4 points)

1. BandMember a = new BandMember();
2. BandMember b = new BandMember("Joey", 24);
3. BandMember c = new BandMember("Johnny", "27");
4. BandMmember d = new BandMember(DeeDee, 24);

a) 1 only
b) 1 and 3 only
c) 1 and 2 only
d) 1 and 4 only
e) 3 only

For an extra 1 point name the band. (The Ramones) (LOL, I won't leave this but I should)

3.3 (20 Points) Create a class called myRect that will be used to create an object that represents a rectangle or square. This class will have methods to calculate area and perimeter of the rectangle . It will contain the following:

1. Two instance variables, width and height.
2. Three constructors – a default with no parameters that sets the rectangle sides to 1, a constructor that takes one integer value and uses it for width and height, creating a square, and a third constructor that takes width and height as parameters.
3. Getters (accessor methods) for both the width and height instance variables.
4. A method called perimeter that returns the perimeter of the rectangle.
5. A method called area that return the area of the rectangle.
6. A toString method that returns the "state" of the rectangle when your object is printed. For example, a rectangle with a width of 3 and a height of 2 would return "a 3x2 rectangle".
7. A main method to test the class. The main method should create three myRect objects, a default rectangle, a 5x5 square and a 3x4 rectangle. Then the code should print the width of the first object, the area of the second and print the third to test your toString method.

Solution

```
public class myRect
{
// A class to keep a myRectangle
private int width;
private int height;

public myRect()
{
    this.width = 1;
    this.height =1;
}

public myRect(int side)
{
    this.width = side;
    this.height =side;
}

public myRect(int w, int h)
{
    this.width = w;
    this.height =h;
}

public int getWidth()
{
```

```java
        return this.width;
    }

public int getHeight()
{
    return this.height;
}

public int perimeter()
{
    return (2*this.width)+(2*this.height);
}

public int area()
{
    return (this.width * this.height);
}

public String toString()
{
    return "A " + this.width + "X" + this.height + " rectangle";
}

// EOC

// test code
public static void main(String[] args)
{
    StdOut.println("My Rectangle");
    myRect R1 = new myRect();
    myRect R2 = new myRect(5);
    myRect R3 = new myRect(3,4);

    StdOut.println(R1.getWidth());
    StdOut.println(R2.area());
    StdOut.print(R3);
}

}
```

Rutgers Honor Pledge

Rutgers honor pledge: *On my honor, I have neither received nor given any unauthorized assistance during this examination.*

You are on your honor to complete this quiz without assistance. You MAY NOT ask any human for assistance or use any online tutoring service.

By continuing, you are AGREEING TO THE ABOVE HONOR STATEMENT and Rutgers Academic Integrity policies.

Problem 4: Sorting & Searching (20 points)

1. **(2.5 points)** If you run the selection sort algorithm on an array of n items, how many times are two array items swapped?

   n-1 times

2. **(2.5 points**) Suppose that selection sort on an array of 100 items has completed 31 iterations of the outer loop. How many items are now guaranteed to be in their final spot (never to be moved again)?

   31

3. (5 points) Mergesort makes two recursive calls.
   a. (2.5 points) What can be said about the array after these recursive calls finish, but before the merge step?

      The array is divided in two sorted regions of roughly n/2 items. If the array has an odd number of items these two regions have sizes n/2 and n/2-1 respectively.

   b. (2.5 points) Show the content of the following array [9, 16, 5, 2, 17, 4, 1] after the two recursive calls finish, but before the merge step.

      [2, 5, 9, 16, 1, 4, 17]

4. (5 points) Suppose that we are sorting an array of eight integers using a quadratic sorting algorithm. After four iterations of the algorithm's outer loop, the array elements are ordered as shown here: 2 4 5 7 8 1 3 6

   Which statement is correct? Note that our selection sort picks largest items first.

   a. The algorithm might be either selection sort or insertion sort.
   b. The algorithm might be selection sort, but it is not insertion sort.
   c. The algorithm is not selection sort, but it might be insertion sort.
   d. The algorithm is neither selection sort nor insertion sort.

14

5. (2.5 points) You are tasked with designing an application that will search an array of 100 items ONCE. Assume that you want the application to have the fastest possible running time, which of the following options would you choose?
    a. Sort the array using mergesort and then search using binary search.
    b. Sort the array using insertion sort and then search using binary search.
    c. Sort the array using selection sort and then search using binary search.
    d. Search the array using sequential search.

6. (2.5 points) You are tasked with designing an application that will search a large array a huge number of times. Assume that you want the application to have the fastest possible running time, which of the following options would you choose?
    a. Sort the array using mergesort and then search using binary search.
    b. Sort the array using insertion sort and then search using binary search.
    c. Sort using selection sort and then search using binary search.
    d. Search the array using sequential search.

Rutgers Honor Pledge

Rutgers honor pledge: *On my honor, I have neither received nor given any unauthorized assistance during this examination.*

You are on your honor to complete this quiz without assistance. You MAY NOT ask any human for assistance or use any online tutoring service.

By continuing, you are AGREEING TO THE ABOVE HONOR STATEMENT and Rutgers Academic Integrity policies.

Problem 5: Complexity (20 points)
For the following ten algorithms, select their Big O Complexity. Select the most efficient Big O (the fastest) for the worse case scenario of the data. Example: If I have an algorithm that adds 1 to every item in an Integer array the Big O is O(n). O(n log n) also works, but it's not needed. O(n) is the fastest.

(for the below choose from: O(1), O(n), O(n2), O(n3), O(n!) or Not Applicable(NA))

1. Find the middle element in an array O(1)
2. Determine if an unsorted array has any duplicates O(n2)
3. Determine if a given array is sorted O(n)
4. Binary search for a target value in an unsorted array NA
5. Convert an array of size n in descending order to an array in ascending order. That is, [3 2 1] -> [1 2 3]
O(n)
6. Given a n x n 2D array of integers, check if there are two elements on the diagonal whose sum is 0.
The diagonal elements in a 2D array are given by a[i][i] (uses the same index for row and column) O(n2)
7. Finding all permutations of the n elements in the array A O(n!)
8. Sort an array of size n using selection sort O(n2)
9. Find the second largest element in an unsorted array O(n)
10. Copy the values from a 2d array to another 2d array (assume they are the same size). O(n2)
(above 1 pt each)

11. The Big O Complexity of a Mergesort is n log n. Explain why. (5 points)
The complexity of Mergesort is n log n because Mergesort is two parts, the splits which are log n for complexity and the merge which is linear as n. In order to get the complexity of the whole sort you need to multiply the two (just like a selection or insertion sorted list is list x list or n2) and you get n log n as a solution

12. If I have an algorithm that runs at n2 how much more time does it take to run my algorithm if I have 4 times as much data as the initial run? 16 times (or two doubling experiments where time is 4 times greater when doubling the data) (5 points)