

**Problem 1-Fix code** Submit a working program called: ArrFix.java

Fix the following code so that the program runs. The comments (in red) tell you what the code **should** be doing.

```
import java.util.ArrayList;
import java.util.Scanner;

public class ArrFix {

    /*This method should return an integer entered by the user*/
    public static void getUserInput()
    {
        Scanner in=new Scanner(System.in);
        System.out.println("Enter a number or -1 to quit.");
        boolean done=true;
        int n=0;

        /*This while loop should run until the user enters a number-once a number is entered, the loop should terminate*/
        while(done)
        {
            try{

                n=in.nextInt();
            }
            catch(Exception e)
            {
                System.out.println("That is not a number. Please enter an number.");
                in.nextLine();
            }
        }

        return done;
    }

    /*This method should take an ArrayList of integers and print out all elements*/
```

```
public static printOutList(a)
{
    System.out.println("Currently in the list:");

    for(int i=0;i<a.size();i++)
    {
        /*Each element is printed out*/
        System.out.printf("%d ", a);
    }

    System.out.println("\n");
}
```

```
public static void main(String [] args){

    /*Declare an ArrayList of integers*/
    ArrayList<Integer> numbers=ArrayList();
    int n;

    do{

        n=getUserInput();

        if(n==-1)
        {
            System.out.println("Bye!");
            return;
        }

        /*Check if the ArrayList already has the number entered by the user*/
        else if(numbers.)
        {
            System.out.println("Already have this number.");
        }

        /*Add the number given by the user to the ArrayList */
        else
        {
```

```
        numbers+n;  
    }
```

```
    printOutList(numbers);
```

```
    }  
    while(n!=-1);
```

```
}
```

```
}
```

### **Sample Run:**

Enter a number or -1 to quit.

4

Currently in the list:

4

Enter a number or -1 to quit.

6

Currently in the list:

4 6

Enter a number or -1 to quit.

cheese

That is not a number. Please enter a number.

donut

That is not a number. Please enter a number.

9

Currently in the list:

4 6 9

Enter a number or -1 to quit.

6

Already have this number.

Currently in the list:

4 6 9

Enter a number or -1 to quit.

-1

Bye!

**Problem 2-Write methods** Submit a program called: `GuessingGame.java`

---

DO NOT MODIFY MAIN. NO CREDIT FOR THE PROGRAM IF MAIN IS MODIFIED.

```
public class GuessingGame {
```

```
    /*Create a method called numGenerate() that randomly generates a number to guess. This method takes an int parameter that serves as a bound (meaning the randomly generated number should not be larger than this bound) and returns the randomly generated number*/
```

```
    /*Create a method called getUserInput(). This method should take a String message that serves as a prompt for the user. The method should return an int that was input by the user.*/
```

```
    /*Create a method called compareGuess(). This method should take two int parameters-one should be the user's guess and the other should be the number the user is trying to guess. The method should return a boolean depending on whether or not the user's guess matches the goal number to be guessed.*/
```

```
    /*Create a method called finalMessage(). This method should take a boolean and an int parameter (this int parameter should represent the number to be guessed). The method simply prints out whether the user won or lost the game and states what the goal number was after. Hint: the boolean parameter determines whether or not the game was won.*/
```

```
    public static void main (String []args) //DO NOT MODIFY.
```

```
    {
        System.out.println("Welcome to the guessing game!");
        System.out.println("You have 3 guesses!");
        int n=getUserInput("Enter the largest number possible to guess (remember, if you want 5 to be the largest enter one larger-6)");

        int numToGuess=numGenerate(n);
        int i=0;

        boolean choice=true;
```

```

while(i<3 && choice)
{
    int userNumber=getUserInput("Enter your guess:");
    choice=compareGuess(userNumber, numToGuess);
    i++;
}
finalMessage(choice, numToGuess);
}

}

```

### **Sample Run:**

Welcome to the guessing game!

You have 3 guesses!

Enter the largest number possible to guess (remember, if you want 5 to be the largest enter one larger-6) 3

Enter your guess:

2

Yes, the number is 2! You win!

### **Problem 3 -Write a program Submit a program called: DecTern.java**

Create a program that converts between decimal (base 10) and ternary (base 3). The ternary base system is composed of the following unique symbols: 0, 1, 2. You may find it helpful to look at class code and the previous HW.

You should, at a minimum, create the following methods:

- 1) A method that intakes a base 10 number and returns a base 3 number.
- 2) A method that intakes a base 3 number and returns a base 10 number.
- 3) A method that gets user input.

**If you do not create these methods, you will not receive any credit for this problem.** You are also welcome to create any helper methods.

The program should have the following functionality:

- 1) The user should be able to enter a number in base 10 or base 3. There should be some sort of indication of the base the number is in so that you know which type of conversion you will be doing.
- 2) The program should then convert the number to the opposite base. For example, if the user entered a number in base 10, the program should convert the number into base 3.

- 3) The program should continue prompting for input and converting until the user enters the word "exit".

**No sample run.**

**Problem 4 -Write methods** Submit a program called: `directMethods.java`

---

1. Create a method called `time()` that takes an integer (total number of seconds) and outputs to screen a clock in 24 hour format (hours, minutes and seconds). The return value should be the number of hours. If the number of seconds entered is greater than the number of seconds in a day, the method should return -1.
2. Create a method called `prime()` that checks if a number is prime or not. It should return true if it is prime and false otherwise.
3. Create a method called `allFactors()` that prints out all the factors of a given number. The return value should be the number of factors.
4. Create a main method using these methods. Your main method should only contain three lines:  
`System.out.println(time(18000));`  
`System.out.println(prime(13));`  
`System.out.println(allFactors(100));`