Guessing game (10 points) Write a guessing game program that asks the user to guess your secret number. If the guess is too high, write a message saying so and ask them to guess again. Similarly, if the guess is too low, write a message saying so and ask them to guess again. If they guess correctly, congratulate them and end the program. You can assume the user will only enter integers.

CODE:

```
import java.util.Scanner;
public class PersonAdressDemo{
public static void main(String[] args) {
        int secretNumber=30;
        Scanner word=new Scanner(System.in);
        System.out.println("Guess a secret number bwetween 1 to 100: ");
        while (true){
            System.out.println("Enter a guess number: ");
            int Guess=word.nextInt();

            if (Guess < secretNumber){
                System.out.println("Too low ,Try again");
            } else if(Guess > secretNumber){
                System.out.println("Too high, Try again");
            } else {
                System.out.println("Congradulation, you have guessed the correct number.");
                break;
            }


        }
    }

}
```

Ans:

```
Guess a secret number bwetween 1 to 100:
Enter a guess number:
1
Too low ,Try again
Enter a guess number:
45
Too high, Try again
Enter a guess number:
30
Congradulation, you have guessed the correct number.
```

Expanding word (5 points)

Write a program that asks the user for a word and prints back out a string in which the first letter of the input word appears once, followed the second letter appearing twice, followed by the third letter appearing three times, etc. For example, the word cat would be output as caattt, the word zebra would be output as zeebbbrrrraaaaa, and the word a would be output as a.

CODE:

```java
import java.util.Scanner;
public class Main{
    public static void main(String[]args) {
        Scanner luck=new Scanner(System.in);
        System.out.print("Enter the word: ");
        String word=luck.nextLine();
        StringBuilder result=new StringBuilder();
        for(int x=0; x< word.length(); x++) {
            char c=word.charAt(x);
            result.append(String.valueOf(c).repeat(x+1));
        }
        System.out.println("Expanded word "+result);

    }
}
```

ANS:

```
Enter the word: MABELE
Expanded word MAABBBEEEELLLLLEEEEEE
```

Pig Latin (10 points) Write a program that asks the user for a sentence and prints the translation of the sentence in the "language" Pig Latin. Each word in the sentence should be translated into Pig Latin in the following way:

1) If the English word starts with one or more consonants, all consonants before the first vowel are removed from the front of the English word, and added to the end of it, keeping the order the same. The string "ay" is then added to the end of this string.

2) If the English word starts with a vowel, append the string "hay" to the end of that word.

Treat 'y' as a consonant if it is at the beginning of a word, and otherwise treat it as a vowel. You may assume that the only punctuation used in the sentence are periods and commas. The displayed sentence should have the same punctuation as the original sentence, but should be all lower-case.

For example, the input hello should be displayed at ellohay, and the sentence The elephant ate a mango. should be displayed at ethay elephanthay atehay ahay angomay

CODE:

```java
import java.util.Scanner;
public class PigLatin {

    public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter a sentence: ");
            String sentence = scanner.nextLine().toLowerCase();
            String[] words = sentence.split(" ");
            StringBuilder translatedSentence = new StringBuilder();

            for (String word : words) {
                String translatedWord = translateWordToPigLatin(word);
                translatedSentence.append(translatedWord).append(" ");
            }

            System.out.println(translatedSentence.toString().trim());
            scanner.close();
        }

        private static String translateWordToPigLatin(String word) {
            if (isVowel(word.charAt(0))) {
                return word + "hay";
            } else {
                int vowelIndex = firstVowelIndex(word);
                return word.substring(vowelIndex) + word.substring(0, vowelIndex) + "ay";
            }
        }

        private static boolean isVowel(char c) {
            return "aeiou".indexOf(c) != -1;
        }

        private static int firstVowelIndex(String word) {
            for (int i = 0; i < word.length(); i++) {
                if (isVowel(word.charAt(i))) {
                    return i;
                }
            }

               return -1;
          }


}
```

ANS:

```
Enter a sentence: Hellow jumanne, how are you doing
ellowhay umanne,jay owhay arehay ouyay oingday
```

Grade Calculator (10 points)

Write a program that asks the user for a list of grades, and returns the average of the grades. You can choose the format in which the grades are entered, but you should communicate this clearly to the user in the program.

For example, if the user enters the grades 89 81 94, the program returns the average 88.

Extra credit (5 points): Modify your program so that the user can enter a weight for each grade. The program then returns the weighted average of the grades.

For example, if the user enters the grades 89 81 94 with the weights 0.25 0.25 0.5, the program returns the weighted average 89.5.

CODE:

```
import java.util.Scanner;
public class GradeCalculator {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the grades separated by spaces: ");
        String[] grades = scanner.nextLine().split(" ");
        int sum = 0;
        for (String grade : grades) {
            sum += Integer.parseInt(grade);
            }
            double average = (double) sum / grades.length;
            System.out.printf("The average grade is %.2f%n", average);
            scanner.close();

    }

}
```

ANS:

```
Enter the grades separated by spaces: 10 56 78 45
The average grade is 47.25
```

MotorBoat Class (25 points)

Implement a class MotorBoat that represents motorboats. A motorboat has attributes for:

- The capacity of the fuel tank
- The amount of fuel in the tank
- The maximum speed of the boat
- The current speed of the boat
- The efficiency of the boat's motor
- The distance traveled

The class has methods to

- constructor
- Return the distance traveled so far If the boat has efficiency e, the amount of fuel used when traveling at a speed s for time t is e x s^2 x t. The distance traveled in that time is s x t. Include a driver program that instantiates several objects of type MotorBoat and demonstrates the class.

CODE:

```
public class MotorBoat {


            private double fuelCapacity;
            private double currentFuel;
            private double maxSpeed;
            private double currentSpeed;
            private double efficiency;
            private double distanceTraveled;

            public MotorBoat(double fuelCapacity, double currentFuel, double maxSpeed, double efficiency) {
                this.fuelCapacity = fuelCapacity;
                this.currentFuel = currentFuel;
                this.maxSpeed = maxSpeed;
                this.efficiency = efficiency;
                this.currentSpeed = 0;
                this.distanceTraveled = 0;
            }

            public void setSpeed(double speed) {
                if (speed <= maxSpeed) {
                    this.currentSpeed = speed;
                } else {
                    System.out.println("Speed exceeds maximum limit.");
                }
            }

            public void travel(double time) {
                double fuelNeeded = efficiency * Math.pow(currentSpeed, 2) * time;
                if (fuelNeeded <= currentFuel) {
                    currentFuel -= fuelNeeded;
                    distanceTraveled += currentSpeed * time;




                } else {
                    System.out.println("Not enough fuel for this trip.");
                }
            }

            public double getDistanceTraveled() {
                return distanceTraveled;
            }

            public static void main(String[] args) {
                MotorBoat boat1 = new MotorBoat(100, 50, 30, 0.1);
                boat1.setSpeed(20);
                boat1.travel(2);
                System.out.println("Boat 1 distance traveled: " + boat1.getDistanceTraveled());

                MotorBoat boat2 = new MotorBoat(150, 100, 40, 0.08);
                boat2.setSpeed(35);
                boat2.travel(1.5);
                System.out.println("Boat 2 distance traveled: " + boat2.getDistanceTraveled());
            }


    }
```

ANS:

```
Not enough fuel for this trip.
Boat 1 distance traveled: 0.0
Not enough fuel for this trip.
Boat 2 distance traveled: 0.0
```

PersonAddress Class (20 points)

Implement a class PersonAddress that represents an entry in an address book. Its attributes are:

- The first name of the person

- The last name of the person
- The email address of the person
- The telephone number of the person It will have methods to
- constructor
- Test whether two instances are equal based solely on name Include a driver program that instantiates several objects of type PersonAddress and demonstrates the class

CODE:

```
public class PersonAddresses {
            private String firstName;
            private String lastName;
            private String email;
            private String phone;

            public PersonAddresses(String firstName, String lastName, String email, String phone) {
                this.firstName = firstName;
                this.lastName = lastName;
                this.email = email;
                this.phone = phone;
            }

            @Override
            public boolean equals(Object obj) {
                if (this == obj) return true;
                if (obj == null || getClass() != obj.getClass()) return false;
                PersonAddresses that = (PersonAddresses) obj;
                return firstName.equals(that.firstName) && lastName.equals(that.lastName);
            }

            public static void main(String[] args) {
                PersonAddresses person1 = new PersonAddresses("mabele", "john", "mabelejohn@example.com", "123-456-7890");
                PersonAddresses person2 = new PersonAddresses("Jane", "Doe", "jane.doe@example.com", "987-654-3210");
                PersonAddresses person3 = new PersonAddresses("mabele", "john", "mabelejohn@example.com", "321-654-9870");

                System.out.println(person1.equals(person2));
                System.out.println(person1.equals(person3));
            }


    }
```

ANS:

```
false
true
```