*7.1 (Assign grades) Write a program that reads student scores, gets the best score, and then assigns grades based on the following scheme:

Grade is A if score is >best - 10

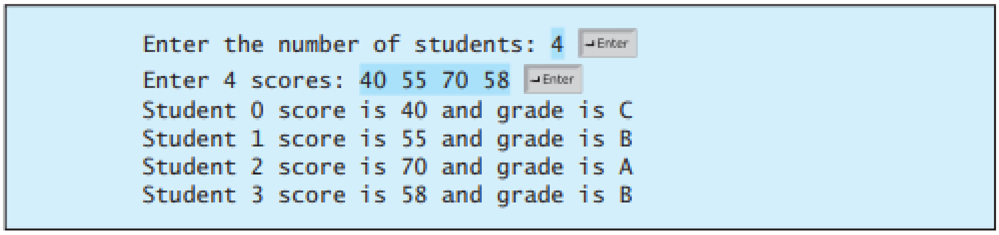
Grade is B if score is > best - 20;

Grade is C if score is > best - 30;

Grade is D if score is > best - 40;

Grade is F otherwise.

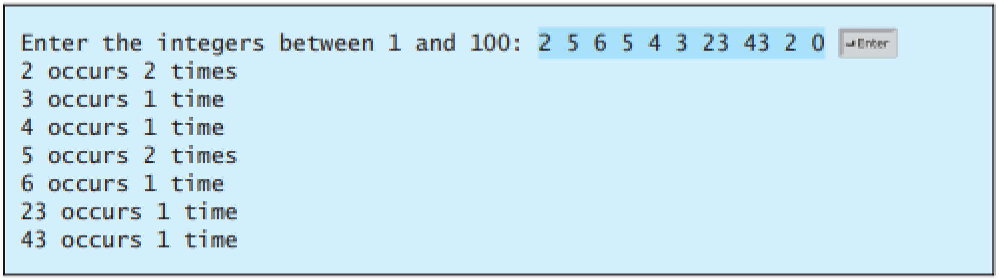
The program prompts the user to enter the total number of students, then prompts the user to enter all of the scores and concludes by displaying the grades. Here is a sample run:



```
Enter the number of students: 4 ↵Enter
Enter 4 scores: 40 55 70 58 ↵Enter
Student 0 score is 40 and grade is C
Student 1 score is 55 and grade is B
Student 2 score is 70 and grade is A
Student 3 score is 58 and grade is B
```

```
package assigngrades;
import java.util.*;
public class AssignGrades {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    int student,best=0;
    String[] grade = {"A","B","C","D","F"};
    System.out.print("Enter the number of student: ");
    student = input.nextInt();
    int[] score = new int[student+1];
    System.out.print("Enter "+student+" scores: ");
    for(int a=0;a<student;a++){
    score[a] = input.nextInt();
    if(score[a] >= best){
      best = score[a];
    }
    }
    for(int a=0;a<student;a++){
    System.out.print("Student "+(a)+" score is "+score[a]+" and grade is ");
    if(score[a]>=(best-10)){
      System.out.print(grade[0]);
    }else{
      if(score[a]>=(best-20)){
        System.out.print(grade[1]);
      }else{
      if(score[a]>=(best-30)){
        System.out.print(grade[2]);
      }else{
      if(score[a]>=(best-40)){
        System.out.print(grade[3]);
      }else{
      if(score[a]<(best-40)){
        System.out.print(grade[4]);
      }}}}}
    System.out.println();
    }
  }
}
```

**7.3 (Count occurrence of numbers) Write a program that reads the integers between 1 and 100 and counts the occurrences of each. Assume the input ends with 0. Here is a sample run of the program:



```
Enter the integers between 1 and 100: 2 5 6 5 4 3 23 43 2 0
2 occurs 2 times
3 occurs 1 time
4 occurs 1 time
5 occurs 2 times
6 occurs 1 time
23 occurs 1 time
43 occurs 1 time
```

```
package countoccurrenceofnumbers;

import java.util.*;
public class CountOccurrenceOfNumbers {

   public static void main(String[] args) {

      Scanner input = new Scanner(System.in);
      int[] integers = new int[100];
      int[] times = new int[100];
      int exit,a=0;
      System.out.print("Enter the integers between 1 anf 100: ");
      do{
         exit = input.nextInt();
         if(exit!=0){
            if(integers[exit]==exit){
               times[exit]++;

            }else{
               integers[exit] = exit;

               times[exit]++;
            }
            a++;
         }
      }while(exit!=0);

      for(int b=0;b<a;b++){
         if(integers[b]!=0){
         System.out.print(integers[b]+" occurs "+times[b]);
         if(times[b]>1){
            System.out.println(" times");
         }else{
            System.out.println(" time");
         }
         }else{
            if(a<100){
            a++;
            }
         }
      }

   }
}
```

7.8 (Average an array) Write two overloaded methods that return the average of an array with the following headers:

public static int average(int[] array)

public static double average(double[] array)

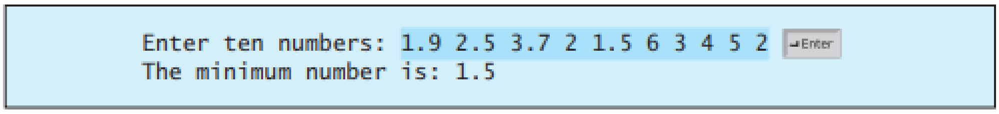
Write a test program that prompts the user to enter ten double values, invokes this method, and displays the average value.

```
package averageanarray;

import java.util.Scanner;
public class AverageAnArray {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        double[] array = new double[10];
        int averageInt;
        double averageDouble = 0;
        System.out.println("Enter 10 double value: ");
        for(int a=0;a<array.length;a++){
            array[a] = input.nextDouble();
        }
        averageInt = (int)average(array);
        averageDouble = average(array);
        System.out.println("Average Int = "+averageInt);
        System.out.println("Average Double = "+averageDouble);
    }

    public static int average(int[] array){
        int average = 0;
        for(int a=0;a<array.length;a++){
        average+=array[a];
        }
        return average/array.length;
    }

    public static double average(double[] array){
        double average = 0;
        for(int a=0;a<array.length;a++){
        average+=array[a];
        }
        return average/array.length;
    }
}
```

7.9 (Find the smallest element) Write a method that finds the smallest element in an array of double values using the following header:

public static double min(double[] array)

Write a test program that prompts the user to enter ten numbers, invokes this method to return the minimum value, and displays the minimum value. Here is a sample run of the program:



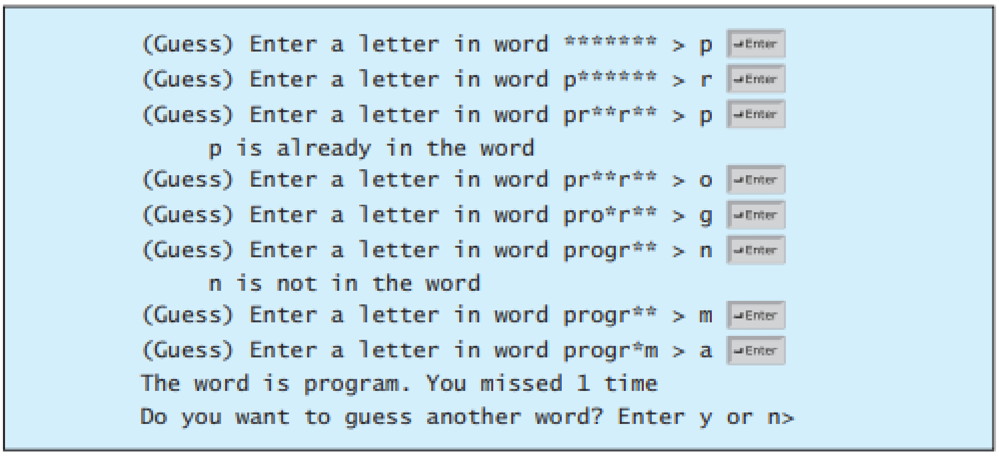
```
Enter ten numbers: 1.9 2.5 3.7 2 1.5 6 3 4 5 2
The minimum number is: 1.5
```

```
package findthesmallestelement;

import java.util.*;

public class FindTheSmallestElement {

   public static void main(String[] args) {

      Scanner input = new Scanner(System.in);
      double[] num = new double[10];
      System.out.print("Enter ten numbers: ");
      for(int a=0;a<num.length;a++){
         num[a] = input.nextDouble();
      }
      System.out.print("The minimum numbers is: "+min(num));
   }

   public static double min(double[] array){
      double min = array[1];
      for(int a=0;a<array.length;a++){
         if(min>=array[a]){
            min = array[a];
         }
      }
      return min;
   }

}
```

***7.35 (Game: hangman) Write a hangman game that randomly generates a word and prompts the user to guess one letter at a time, as shown in the sample run. Each letter in the word is displayed as an asterisk. When the user makes a correct guess, the actual letter is then displayed. When the user finishes a word, display the number of misses and ask the user whether to continue to play with another word. Declare an array to store words, as follows:

// Add any words you wish in this array

String[] words = {"write", "that", ...};



```
(Guess) Enter a letter in word ******* > p
(Guess) Enter a letter in word p****** > r
(Guess) Enter a letter in word pr**r** > p
     p is already in the word
(Guess) Enter a letter in word pr**r** > o
(Guess) Enter a letter in word pro*r** > g
(Guess) Enter a letter in word progr** > n
     n is not in the word
(Guess) Enter a letter in word progr** > m
(Guess) Enter a letter in word progr*m > a
The word is program. You missed 1 time
Do you want to guess another word? Enter y or n>
```

```
package gamehangman;

import java.util.*;

public class GameHangman {

  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

                String play;
                do {
                        char[] word = getWord();

                        // Display each letter in the word as an asterisk.
                        char[] asterisks = new char[word.length];
                        fillAsterisks(asterisks);

                        int missed = 0;
                        do {
                                char guess = getGuess(asterisks);

                                if (!isCorrectGuess(word, asterisks, guess))
                                        missed++;

                        } while (!isWordFinish(asterisks));

                        print(word, missed);

                        System.out.println("Do you want to guess another word? Enter y or n>");
                        play = input.next();

                } while (play.charAt(0) == 'y');
        }

        public static char[] getWord() {
```

```
        String[] words = {"write", "that", "program", "monkey", "rooster", "dog",
                                    "pig", "rat", "ox", "tiger",
"rabbit", "dragon", "snake",
                                    "horse", "sheep"};

        String pick = words[(int)(Math.random() * words.length)];
        char[] word = new char[pick.length()];

        for (int i = 0; i < word.length; i++) {
            word[i] = pick.charAt(i);
        }
        return word;
    }

    public static void fillAsterisks(char[] list) {
        for (int i = 0; i < list.length; i++) {
            list[i] = '*';
        }
    }

    public static boolean isCorrectGuess(char[] word, char[] blanks, char guess) {
        boolean correct = false;
        int message = 2;
        for (int i = 0; i < word.length; i++) {
            if (word[i] == guess) {
                correct = true;
                if (blanks[i] == guess)
                    message = 1;
                else {
                    blanks[i] = guess;
                    message = 0;
                }
            }
        }
        if (message > 0)
            print(message, guess);
        return correct;
    }

    public static boolean isWordFinish(char[] blanks) {
        for (char e: blanks) {
            if (e == '*')
                return false;
        }
        return true;
    }

    public static void print(char[] word, int missed) {
        System.out.print("The word is ");
        System.out.print(word);
        System.out.println(" You missed " + missed + " time");
    }

    public static void print(int m, char guess) {
        System.out.print("\t" + guess);
        switch (m) {
            case 1 : System.out.println(" is already in the word"); break;
            case 2 : System.out.println(" is not in the word");
        }
    }

    public static char getGuess(char[] asterisks){
        Scanner input = new Scanner(System.in);
        System.out.print("(Guess) Enter a letter in word ");
        System.out.print(asterisks);
        System.out.print(" > ");
        String g = input.next();
        return g.charAt(0);
    }
}
```