Problem 1:

Write a program that reads a set of floating-point values. Ask the user to enter the values, then print:

The average of the values

The smallest of the values

The largest of the values

The range, that is the difference between the smallest and the largest

Of course, you may only prompt for the values once.

|  |
| --- |
| **import** java.util.Scanner;  /\*  Name: Casey Carnnia  Date: 10.22.2012  Scope: Problem 1: Write a program that reads a set of floating-point values. Ask the user to enter the values, then print:  The average of the values  The smallest of the values  The largest of the values  The range, that is the difference between the smallest and the largest Of course, you may only prompt for the values once.  Solution: prompt the user for input.  \*\*Important\*\*  the problem says a set of numbers so we don't need to program for a variable set. we can say a set of 5, or 10.  Instantiate an array with a length of 5  using a loop  Instantiate an counter  as long as that counter is smaller then 5  read the user input into next index of the array    instantiate a variable sum = 0 type = double  instantiate a variable averageOutput = 0 type = double  using a loop  instantiate a counter  as long as the counter is less then the length of the array  add value of this index of array to sum  averageOutput is sum divided by length of array  output the averageOutput  instantiate a variable maxValue = 0 type = double  using a loop  instantiate a counter  as long as the counter is less then array length  if this index of array is greater then maxValue  set maxValue to this instance of array  output the maxValue      instantiate a variable minValue = 0 type = double  using a loop  instantiate a counter  as long as the counter is less then array length  if this index of array is less then minValue  set minValue to this instance of array  output the minValue    out put the range as maxValue - minValue        \*/    // declare class  **public** **class** AveragesForLoops{  //declare the main method  **public** **static** **void** main(String[] args) {    // welcome the user  System.*out*.println("I can calculate the average of a bunch of numbers and also tel you the larges and the smalles of the numbers and their range.");  // get ready to read the user data  Scanner keyboard = **new** Scanner(System.*in*);  // prompt the user for input  // store the user input  System.*out*.print("Please enter a set of 5 numbers seperated by blank space. It is OK to use fractions:");  **double**[] inputArray;  inputArray = **new** **double**[5];    **for** (**int** inputCounter = 0; inputCounter < 5; inputCounter ++){  inputArray[inputCounter] = keyboard.nextDouble();  }//input loop END    **double** sum = 0;  **for** (**int** averageCounter = 0; averageCounter < inputArray.length; averageCounter ++){  sum = sum + inputArray[averageCounter];  }//average loop END  **double** averageOutput = sum / inputArray.length;  System.*out*.printf("The average if the numbers you entered is: %10.2f", averageOutput);  System.*out*.print("\n");    **double** maxValue = inputArray[0];  **for** (**int** maxCounter = 0; maxCounter < inputArray.length; maxCounter ++){  **if** (inputArray[maxCounter] > maxValue){  maxValue = inputArray[maxCounter];  }//if END  }//MAX loop END  System.*out*.printf("The largest value you entered is: %10.2f", maxValue);  System.*out*.print("\n");    **double** minValue = inputArray[0];  **for** (**int** minCounter = 0; minCounter < inputArray.length; minCounter ++){  **if** (inputArray[minCounter] < minValue){  minValue = inputArray[minCounter];  }//if END  }//MIN loop END  System.*out*.printf("The smallest value you entered is: %10.2f", minValue);  System.*out*.print("\n");    System.*out*.printf("The range of values you entered is: %10.2f", (maxValue - minValue));  System.*out*.print("\n");        }//main END  }//class END |
|  |

Problem 2:

Write a program that reads a word and prints each character of the word on a separate line. For example, if the user provides the input "Harry", the program prints:

H

a

r

r

y

|  |
| --- |
| **import** java.util.Scanner;  /\*  Name: Casey Carnnia  Date: 10.22.2012  Scope: Problem 2: Write a program that reads a word and prints each character of the word on a separate line.  For example, if the user provides the input "Harry", the program prints:  H  a  r  r  y  Solution: welcome the user  instantiate a variable userInput = "" type = String  prompt the user for input  store the user input in userInput  using a loop  instantiate a counter  as long as the counter is less then the length of userInput  output using println() this index of userInput  \*/    // declare class  **public** **class** StringForLoops{  //declare the main method  **public** **static** **void** main(String[] args) {    // welcome the user  System.*out*.println("I loop throughn the charecters of an string. ");  // get ready to read the user data  Scanner keyboard = **new** Scanner(System.*in*);  // prompt the user for input  // store the user input  System.*out*.print("Please enter a word or a name:");  String userInput;  userInput = keyboard.next();  **for** (**int** inputCounter = 0; inputCounter < userInput.length(); inputCounter ++){  System.*out*.println(userInput.charAt(inputCounter));  }//input loop END      }//main END  }//class END |
|  |

Problem 3:

The Fibonacci numbers are defined by the sequence:

f1 = 1

f2 = 1

fn = fn-1 + fn-2.

Reformulate that as Fold1 = 1;

Fold2=1;

Fnew=fold1+fold2;

After that, discard fold2, which is no longer needed, and set fold2 to fold1, and fold1 to fnew. Repeat an appropriate number of times.

Implement a program that prompts the user for an integer n and prints the nth Fibonacci number, using the above algorithm.

|  |
| --- |
| **import** java.util.Scanner;  /\*  Name: Casey Carnnia  Date: 10.22.2012  Scope: Problem 3: The Fibonacci numbers are defined by the sequence:  f1 = 1  f2 = 1  fn = fn-1 + fn-2.  Reformulate that as  Fold1 = 1;  Fold2=1;  Fnew=fold1+fold2;  After that, discard fold2, which is no longer needed, and set fold2 to fold1, and fold1 to fnew. Repeat an appropriate number of times.  Implement a program that prompts the user for an integer n and prints the nth Fibonacci number, using the above algorithm.  Solution: welcome the user  instantiate a variable userInput = "" type = int  prompt the user for input  store the user input in userInput  using a loop  instantiate a counter  as long as the counter is less then the userInput  foldNew = total of fold1 and fold2  fold2 = fold1  fold1 = foldNew    output using println() foldNew  \*/    // declare class  **public** **class** Fibonacci{  //declare the main method  **public** **static** **void** main(String[] args) {    // welcome the user  System.*out*.println("I can tell you the Nth Fibonacci number.");  // get ready to read the user data  Scanner keyboard = **new** Scanner(System.*in*);  // prompt the user for input  // store the user input  System.*out*.print("Please enter an integer to test this program:");  **int** userInput;  userInput = keyboard.nextInt();  **int** fold1 = 0;  **int** fold2 = 1;  **int** foldNew = 0;  **int** inputCounter;    **for** ( inputCounter = 0; inputCounter < userInput; inputCounter ++){  foldNew = fold1 + fold2;  fold2 = fold1;  fold1 = foldNew;  }//input loop END    System.*out*.println("The" + inputCounter + "th Fibonacci is: " + foldNew);    }//main END  }//class END |
|  |

Problem 4:

Write a program that prints a multiplication table, like this:

1 2 3 4 5 6 7 8 9 10

2 4 6 8 10 12 14 16 18 20

3 6 9 12 15 18 21 24 27 30

…

10 20 30 40 50 60 70 80 90 100

|  |
| --- |
| **import** java.util.Scanner;  /\*  Name: Casey Carnnia  Date: 10.22.2012  Scope: Problem 4: Write a program that prints a multiplication table, like this:  1 2 3 4 5 6 7 8 9 10  2 4 6 8 10 12 14 16 18 20  3 6 9 12 15 18 21 24 27 30  …  10 20 30 40 50 60 70 80 90 100  Solution: welcome the user  using a loop  instantiate a counter  as long as this loops counter is less then or equal to 10  using another loop  as long as this loops counter is less then or equal to 10  output outer loop counter times inner loop counter separated by tabs  output a line brake  \*/    // declare class  **public** **class** multiplicationTable{  //declare the main method  **public** **static** **void** main(String[] args) {    // welcome the user  System.*out*.println("I can print a multiplication table.");    **for** (**int** outterCounter = 1; outterCounter <= 10; outterCounter ++){  **for** (**int** innerCounter = 1; innerCounter <= 10; innerCounter ++){  System.*out*.print(outterCounter \* innerCounter + "\t");  }//inner loop END  System.*out*.print("\n");  }//outter loop END      }//main END  }//class END |
|  |

Problem 5:

The Drunkard's Walk. A drunkard in a grid of streets randomly picks one of four directions and stumbles to the next intersection, then again randomly picks one of the four directions, and so on. You might think that on average the drunkard doesn't move very far because the choices cancel each other out, but that is actually not the case.

Represent locations as integer pairs(x,y). Implement the drunkard's walk over 100 intersections, starting at (0,0) and print the ending location.

|  |
| --- |
| **import** java.util.Random;  **import** java.lang.Math;  /\*  Name: Casey Carnnia  Date: 10.22.2012  Scope: Problem 5: The Drunkard's Walk.  A drunkard in a grid of streets randomly picks one of four directions and stumbles to the next intersection,  then again randomly picks one of the four directions, and so on.  You might think that on average the drunkard doesn't move very far because the choices cancel each other out,  but that is actually not the case.  Represent locations as integer pairs(x,y). Implement the drunkard's walk over 100 intersections,  starting at (0,0) and print the ending location.  Solution: welcome the user  instantiate a variable x = 0 type = int  instantiate a variable y = 0 type = int  instantiate a variable distance = 0 type = double  using a loop  initialize a counter  as long as the counter is less then 100  pick random number between 1 and 4  if 1 we are going east so increment y by 1  if 2 we are going west so decrement y by 1  if 3 we are going north so increment x by 1  if 4 we are going south so decrement x by 1    use since we started from position (0,0):  modify the distance formula so distance = sqrt of [(x - 0)^2 + (y - 0)^2]  output the answer  \*/    // declare class  **public** **class** drunkardWalk{  //declare the main method  **public** **static** **void** main(String[] args) {    // welcome the user  System.*out*.println("If a drunk person at location (x,y) randomly travels i various directions in a grid city. What happens after 100 random choices.");  Random randomGenerator = **new** Random();  **int** x = 0;  **int** y = 0;  **double** distance = 0;  **for** (**int** decisionCounter = 0; decisionCounter < 100; decisionCounter ++){  **int** direction = randomGenerator.nextInt(4) + 1 ;  **switch** (direction) {  **case** 1: x = x + 1; //going east one block  **break**;  **case** 2: x = x + -1;//going west one block  **break**;    **case** 3: y = y + 1;//going north one block  **break**;    **case** 4: y = y - 1;//going south one block\  **break**;  }//switch END    }//loop END  // d= sqrt of [(x - 0)^2 + (y - 0)^2]  distance = Math.*sqrt*(((x \* x) + (y \* y)));        System.*out*.printf("If the city blockes are one mile by one mile then the drunk person has traveled %4.2f", distance);  System.*out*.println(" miles away from his original location by walking 100 miles. ");      }//main END  }//class END |
|  |