**Exercise 1 - Dawson**

import java.util.Scanner;

public class CharsInLine

{

public static void main(String[] args)

{

Scanner console = new Scanner(System.in);

int numberUpperCase = 0; // keeps track of the number of uppercase

int numberLowerCase = 0; // characters that have been read

int numberOfDigits = 0;

int numberWhiteSpace = 0;

// issue prompt for getting user to supply text

System.out.print("Enter a line of text : ");

String inLine = console.nextLine();

int i = 0;

while ( i < inLine.length() ) {

// process the ith character in the input line

char currentCharacter = inLine.charAt(i);

// test whether the character is uppercase

if ( Character.isUpperCase(currentCharacter) ) {

// got another uppercase character

numberUpperCase++;

}

if(Character.isDigit(currentCharacter)) {

numberOfDigits++;

}

if(Character.isLowerCase(currentCharacter)) {

numberLowerCase++;

}

if(Character.isWhitespace(currentCharacter)) {

numberWhiteSpace++;

}

// current character has been processed

i++;

// ready to repeat the inner loop to process next character

}

System.out.println("There are " + numberUpperCase + " upper case characters" );

System.out.println("There are " + numberLowerCase + " lower case characters" );

System.out.println("There are " + numberOfDigits + " digits" );

System.out.println("There are " + numberWhiteSpace + " white space characters" );

console.close();

}

}

**Output - Exercise 1**

Enter a line of text : The University of Southern Maine congratulates the more than 1,000 students who met the criteria to be placed on the 2017 Fall Semester Dean's List.

There are 8 upper case characters

There are 105 lower case characters

There are 8 digits

There are 24 white space characters

**Exercise 2 - Dawson**

import java.util.Random;

public class RandomWalk {

public static void main(String[] args) {

Random rand = new Random();

int position = 0;

while (position <= 10 && position >= -10){

if (rand.nextDouble() < 0.5){

position--;

}else{

position++;

}

System.out.print(" " + position); // print each position

}

System.out.println(); // print newline at end of position list

System.out.println("Final position is : " + position);

}

}

**Output - Exercise 2**

-1 -2 -1 0 1 0 1 0 1 2 3 4 5 6 7 8 7 6 5 4 3 2 1 2 1 0 1 2 1 2 1 0 1 0 -1 -2 -3 -4 -3 -2 -1 -2 -1 -2 -3 -2 -1 0 1 2 3 4 3 4 3 2 1 0 1 0 -1 0 1 0 -1 0 1 2 3 2 3 4 5 4 3 2 1 0 1 0 -1 -2 -1 0 -1 0 1 0 -1 -2 -3 -4 -3 -4 -5 -6 -5 -6 -5 -6 -5 -4 -3 -2 -1 -2 -3 -2 -3 -2 -3 -4 -3 -4 -5 -6 -5 -6 -7 -8 -9 -8 -7 -8 -7 -6 -5 -6 -5 -4 -5 -6 -5 -6 -7 -6 -7 -6 -7 -8 -9 -8 -9 -8 -9 -8 -7 -6 -7 -6 -7 -6 -7 -8 -7 -8 -7 -8 -7 -8 -9 -8 -9 -10

Final position is : -10

-1 -2 -1 -2 -1 0 1 0 -1 0 1 0 1 0 -1 0 1 0 1 0 1 0 -1 -2 -1 -2 -3 -2 -1 0 -1 0 -1 -2 -3 -2 -1 0 -1 0 -1 -2 -3 -4 -3 -4 -3 -2 -1 -2 -3 -2 -1 0 1 2 3 2 1 0 -1 -2 -1 -2 -1 0 -1 -2 -1 0 -1 0 1 0 1 2 3 2 1 2 3 4 5 6 5 6 7 6 5 4 5 6 5 6 7 8 7 6 7 8 7 8 7 8 9 8 9 10

Final position is : 10

**Exercise 3 - Dawson**

import java.util.Random;

public class RandomWalk {

public static void main(String[] args) {

Random rand = new Random();

int position = 0;

int steps = 0;

int zeroCross = 0;

while (position < 10 && position > -10){

if (rand.nextDouble() < 0.5){

position--;

}else{

position++;

}

if(position == 0) {

zeroCross++;

}

System.out.print(" " + position); // print each position

steps++;

}

System.out.println(); // print newline at end of position list

System.out.println("Final position is : " + position);

System.out.println("Total number of steps was: " + steps);

System.out.println("Times crossed zero: " + zeroCross);

}

}

**Output - Exercise 3**

1 0 -1 -2 -1 0 1 0 -1 0 1 0 1 2 3 4 3 4 3 4 5 4 3 4 3 2 3 2 3 4 5 4 5 6 7 8 9 8 9 8 9 8 9 8 9 8 7 8 7 8 9 10

Final position is : 10

Total number of steps was: 52

Times crossed zero: 5

-1 0 1 2 1 0 -1 -2 -3 -2 -3 -4 -5 -6 -7 -6 -7 -8 -9 -10

Final position is : -10

Total number of steps was: 20

Times crossed zero: 2

**Exercise 4 - Dawson**

import java.util.Random;

public class RandomWalk4 {

public static void main(String[] args) {

Random rand = new Random();

int position = 0;

int steps = 0;

int zeroCross = 0;

while (position < 10 && position > -10 && steps < 50){

if (rand.nextDouble() < 0.5){

position--;

}else{

position++;

}

if(position == 0) {

zeroCross++;

}

System.out.print(" " + position); // print each position

steps++;

}

System.out.println(); // print newline at end of position list

System.out.println("Final position is : " + position);

System.out.println("Total number of steps was: " + steps);

System.out.println("Times crossed zero: " + zeroCross);

}

}

**Output - Exercise 4**

1 0 1 2 3 2 1 0 -1 0 1 0 1 2 1 0 -1 -2 -3 -2 -1 -2 -3 -4 -3 -2 -3 -2 -1 -2 -1 -2 -1 0 -1 0 -1 0 -1 -2 -3 -2 -1 -2 -1 -2 -3 -4 -3 -2

Final position is : -2

Total number of steps was: 50

Times crossed zero: 8

-1 0 1 2 3 2 1 2 1 0 -1 0 -1 0 1 2 3 4 3 2 3 2 1 2 3 2 3 4 5 4 5 4 3 2 1 0 1 2 1 0 1 0 -1 0 -1 -2 -1 0 1 0

Final position is : 0

Total number of steps was: 50

Times crossed zero: 10

**Exercise 5 - Dawson**

import java.util.Random;

public class RandomWalk4 {

public static void main(String[] args) {

//initialize variables

int position, steps, loops = 0, endTen = 0, endNTen = 0, endSteps = 0, p1, p2,p3;

Random rand = new Random();

//start simulation loop

for(int i = 0 ; i<1000; i++) {

//reset position and steps from last simulation

position = 0;

steps = 0;

//simulation

while (position < 10 && position > -10 && steps < 50){

if (rand.nextDouble() < 0.5){

position--;

}else{

position++;

}

steps++;

}

//log result before next simulation

if(position == 10) {

endTen++;

}else if(position == -10) {

endNTen++;

}else {

endSteps++;

}

loops = i;

}

//calculate approximate % values

p1 = Math.round(((float) endTen / (loops + 1))\*100);

p2 = Math.round(((float) endNTen / (loops + 1))\*100);

p3 = Math.round(((float) endSteps / (loops + 1))\*100);

//print statistics

System.out.println("Number of cycles run: " + (loops + 1));

System.out.println("Ended at 10 : " + endTen+ "/" + (loops + 1) + " or about " + p1 + "%");

System.out.println("Ended at -10 : " + endNTen+ "/" + (loops + 1) + " or about " + p2 + "%");

System.out.println("Ended at 50 steps : " + endSteps+ "/" + (loops + 1) + " or about " + p3 + "%");

}

}

**Output - Exercise 5**

Number of cycles run: 1000

Ended at 10 : 144/1000 or about 14%

Ended at -10 : 169/1000 or about 17%

Ended at 50 steps : 687/1000 or about 69%

Number of cycles run: 1000

Ended at 10 : 153/1000 or about 15%

Ended at -10 : 142/1000 or about 14%

Ended at 50 steps : 705/1000 or about 71%

Number of cycles run: 1000

Ended at 10 : 171/1000 or about 17%

Ended at -10 : 185/1000 or about 19%

Ended at 50 steps : 644/1000 or about 64%