**9: Implementation of Inverse transform technique for generating random variates from Uniform/weibull/exponential/Triangular distribution.**

**Program:**

import java.util.Scanner;

class RV {

static double[] generatern() {

Scanner s = new Scanner(System.in);

System.out.print("For generating random numbers using LCM");

System.out.print("\nseed:");

int seed = s.nextInt();

System.out.print("a:");

int a = s.nextInt();

System.out.print("m:");

int m = s.nextInt();

System.out.print("c:");

int c = s.nextInt();

System.out.print("No. of random numbers you want:");

int n = s.nextInt();

double rn[] = new double[n];

rn[0] = (a \* seed + c) % m;

rn[0] /= 10;

System.out.println("\nGenerating random numbers:");

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

if (i < n - 1) {

rn[i + 1] = (a \* rn[i] \* 10 + c) % m;

rn[i + 1] /= 10;

}

System.out.print(rn[i] + " ");

}

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

return rn;

}

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

double rn[] = generatern();

System.out.println("Generate random variates for");

System.out.println("\n1.Uniform\n2.Exponential\n3.Triangular\n4.Weibull\n5.Stop");

int choice = s.nextInt();

while (choice != 5) {

switch (choice) {

case 1:

uniform(rn);

break;

case 2:

expo(rn);

break;

case 3:

triangular(rn);

break;

case 4:

weibull(rn);

case 5:

break;

default:

System.out.println("Invalid choice");

}

System.out.println("\n1.Uniform\n2.Exponential\n3.Triangular\n4.Weibull\n5.Stop");

choice = s.nextInt();

}

}

static void expo(double[] rn) {

Scanner s = new Scanner(System.in);

System.out.print("lambda:");

double l = s.nextInt();

System.out.println("Random variates:");

for (int i = 0; i < rn.length; i++) {

System.out.print((-Math.log(1 - rn[i]) / l) + " ");

}

System.out.println("");

}

static void uniform(double[] rn) {

Scanner s = new Scanner(System.in);

System.out.print("a:");

int a = s.nextInt();

System.out.print("b:");

int b = s.nextInt();

System.out.println("Random variates:");

for (int i = 0; i < rn.length; i++) {

System.out.print((rn[i] \* (b - a) + a) + " ");

}

System.out.println("");

}

static void weibull(double[] rn) {

Scanner s = new Scanner(System.in);

System.out.print("v:");

double v = s.nextDouble();

System.out.print("alpha:");

double a = s.nextDouble();

System.out.print("beta:");

double b = s.nextDouble();

b = 1.0 / b;

System.out.println("Random variates:");

for (int i = 0; i < rn.length; i++) {

System.out.print(((Math.pow(-Math.log(1 - rn[i]), b) \* a)+v) + " ");

}

System.out.println("");

}

static void triangular(double[] rn) {

Scanner s = new Scanner(System.in);

System.out.print("a:");

double a=s.nextDouble();

System.out.print("b:");

double b=s.nextDouble();

System.out.print("c:");

double c=s.nextDouble();

System.out.println("Random variates:");

for (int i = 0; i < rn.length; i++) {

if(0<rn[i]&&(b-a)/(c-a)>=rn[i])

System.out.print(Math.pow(rn[i]\*(b-a)\*(c-a),0.5)+a+" ");

else

System.out.print(c-Math.pow((1-rn[i])\*(c-b)\*(c-a),0.5)+" ");

}

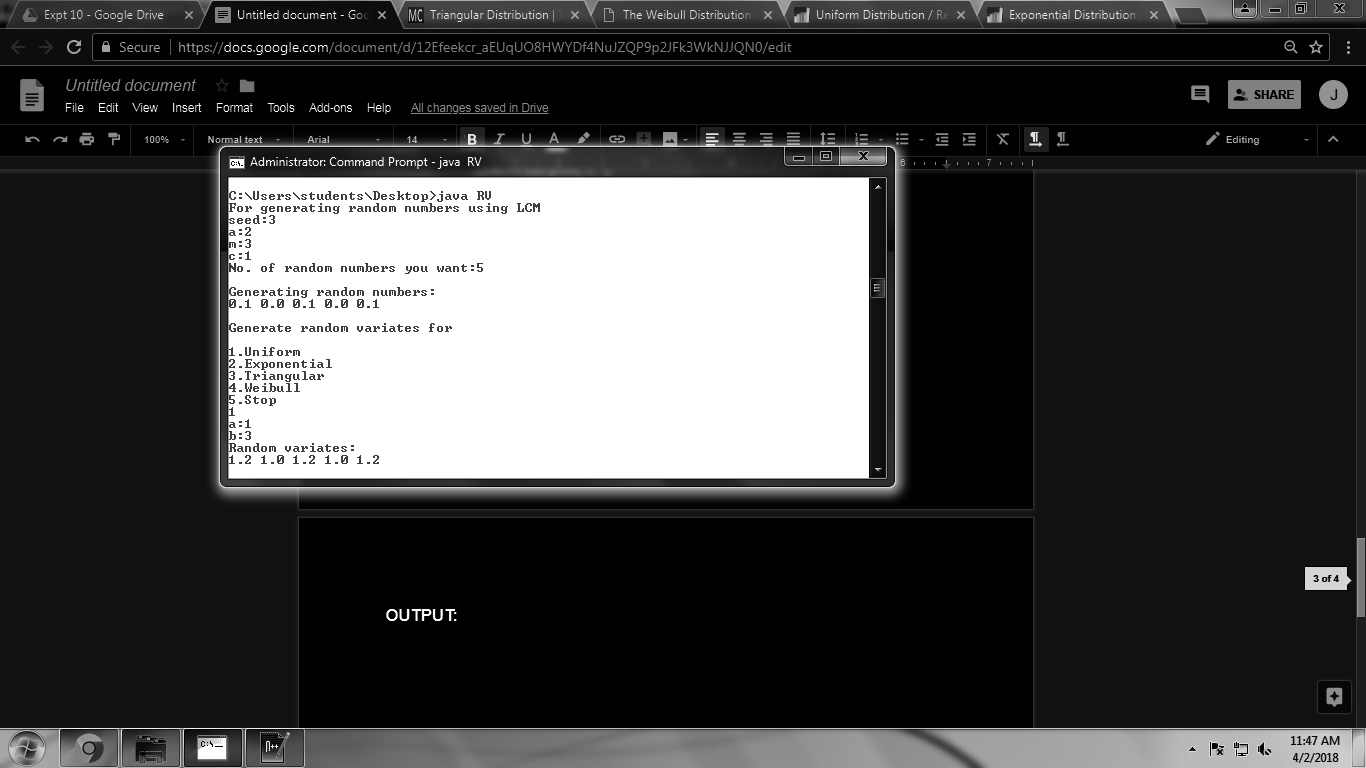
System.out.println("");

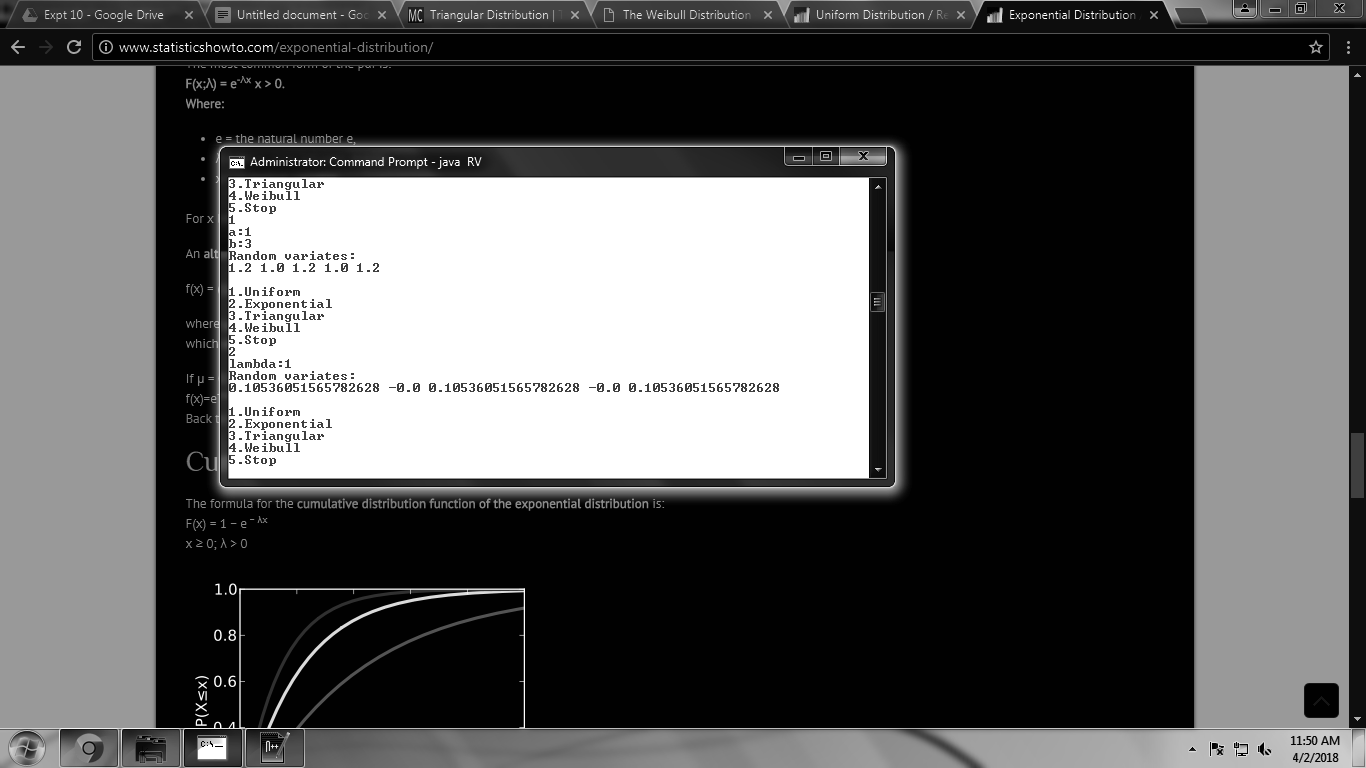
}

}

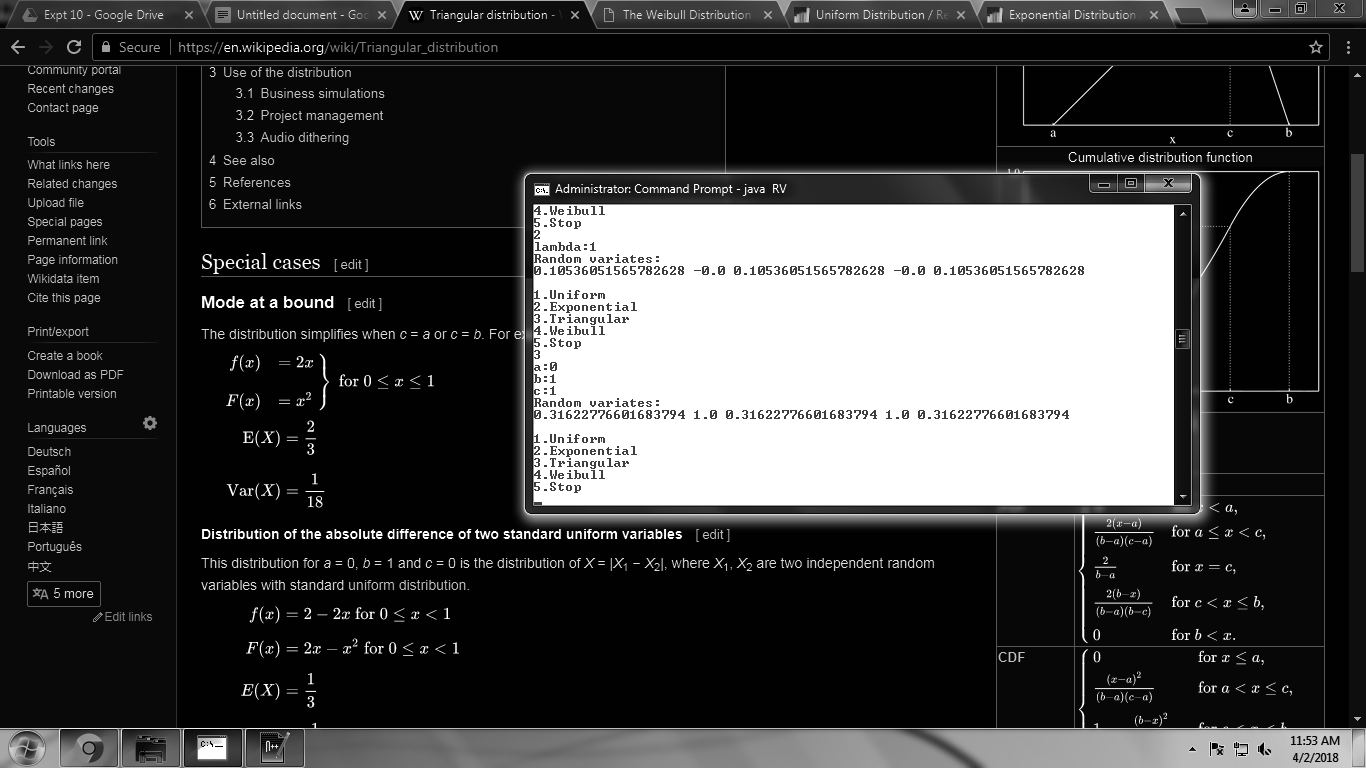
**OUTPUT:**

**Uniform Distribution**

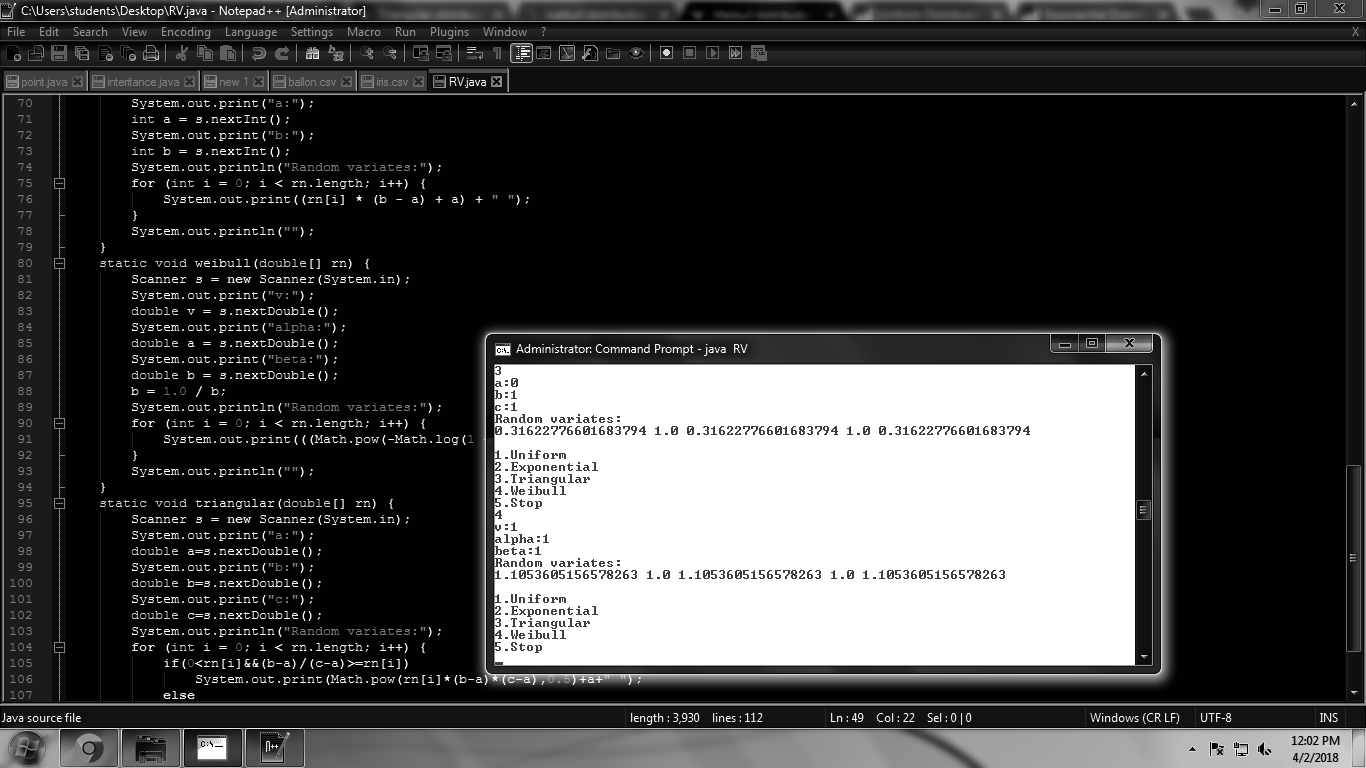
****

**Exponential Distribution**

**Triangular Distribution**

****

**Weibull Distribution**

****