Java Code:

/\*

\* Title: Fermat's Last Theorem, Find Near Miss

\* 2 libraries are used, 1st one for taking user input, 2nd one for power function

\* No extra or external files are used

\*

\* Group member: Sankar Vinay Kumar Chintalapudi, SankarVinayKumarCh@lewisu.edu

\* Group member: Gopi Sandeep Yerra, gopisandeepyerra@lewisu.edu

\*

\* Here we are solve x^n + y^n <> z^n

\*

\* and finding the near miss value and relative miss

\*

\* and then we are also finding the nearest miss

\*/

**import** java.util.Scanner; // Scanner library to take the user input

**import** java.lang.Math; // Math library to take the power of the value

//Main Class where we are taking user inputs setting the criteria and solving the equation

**public** **class** Main {

**public** **static** **void** main(String[] args)

{

**int** x=0,y=0,z=0,n=0;

**double** nearestmiss = 1000; // when we try to find minimum number then we initialize the variable with largest number

**int** initialk = 10; // lower limit for K

**int** finalk = 50; // upper limit for K

Scanner obj = **new** Scanner(System.***in***); // Create a Scanner object

System.***out***.print("How many times do you want to find the near miss: ");

**int** iterations = obj.nextInt(); //Number of times equation will be solved

**for** (**int** i =0; i<iterations; i++) // equation will be solved for iterations times

{

**boolean** check = **false**;

// while loop to maintain the correct values of the variables

**while**(check!=**true**)

{

System.***out***.print("Enter the vlaue of x : ");

x = obj.nextInt();

System.***out***.print("Enter the vlaue of y : ");

y = obj.nextInt();

System.***out***.print("Enter the vlaue of n : ");

n = obj.nextInt();

// check if values of x and y are in range

**if**((x>=initialk && x<=finalk)&&(y>=initialk && y<=finalk)&&(n>2&&n<12))

{

check = **true**;

}

}

//calculating x power n

**int** xn = (**int**) (Math.*pow* (x, n));

//calculating y power n

**int** yn = (**int**) (Math.*pow* (y, n));

//x power n + y power n

**int** xn\_plus\_yn = xn+yn;

//calculating z, by taking n root

z = (**int**) Math.*pow* (xn\_plus\_yn, 1.0/n);

//calculating near miss

**int** nearmiss = (**int**)(xn\_plus\_yn - Math.*pow* (z, n));

//calculating relative miss

**double** relativemiss = 100. \* nearmiss / xn\_plus\_yn;

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

System.***out***.println(" Near Miss: "+nearmiss+" Relative Miss: "+relativemiss);

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

//this check will maintain the lowest value, in other words nearest miss

**if**(relativemiss<nearestmiss)

{

nearestmiss=relativemiss;

}

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

System.***out***.println(" \*\*\*\*\* Nearest Miss: "+nearestmiss+" \*\*\*\*\*");

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

}

}

}



