CS431: Assignment 3
Deadline: 11.55PM 18 September 2013

Submission Procedure: Email your prolog program and test cases as attachment to < asahu AT iitg.ernet.in> and name of the attached file should be #RollNo.Assign3.CS431.pl
Simple but efficient Rule: Copy case Lead to F grade.

Assignment Statement:

Obtain the least square polynomial (straight line) approximation for any given **f(x)** in the range **[a, b]** for **N** given points (**a, a+h, a+2h, …and b**) where **h=(b-a)/N**.

Let $\mathbf{P(x)=c_0+c_1x}$ be a straight-line least square approximation, then this satisfy

$$\mathbf{c_0\,(N+1) \;+\; c_1 \sum x_i \;=\; \sum f(x_i)}$$

$$\mathbf{c_0 \sum x_i \;+\; c_1 \sum x_i^{\,2} \;=\; \sum x_i\, f(x_i)}$$

We can find this co-efficient ($\mathbf{c_0}$ **and** $\mathbf{c_1}$) by Cramer's rule of solving one degree equation. (Cramer's rule: if **ax+by=e** and **cx+dy=f**, then **x=(ed-bf)/(ad-bc)** and **y = (af-ec)/(ad-bc)** when **(ad-bc)≠0**).

List square error (LSE) can be computed as $\sum \left|\, \mathbf{f(x_i) - (c_0+c_1x_i)} \right|^2$

Implement LSE approximation in prolog and approximate and estimation LSE for some your function $\mathbf{x^3+1}$, $\mathbf{x^2}$, x+1 and so on.

Output: c_o, c_1 and LSE
Input : **fn**, a, b, N
/* *you may not be able to pass function as argument, so you can create separate query for separate input test function**/
Hint: Create list of data points, calculate sum, square sum, …