CS431 : Assignment 1
Deadline :11.55PM 15 Aug 2013

Submission Procedure: Email your ML program and test cases as attachment to < asahu AT iitg.ernet.in> and name of the attached file should be #RollNo.Assign1.CS431.ml

Copy case Lead to F grade: We have software to detect copy case; it can detect copy with changed variable and displacement of code.

Assignment Statement:

Implement simpson's rule for Integration in SML and try out some your function on polynomials x^3+1, x^2 , x+1 and so on.

Simpson rule is a more accurate way to integrate function numerically then trapezoidal rule (***discussed in CS431 Tutorial 1***). If we evaluate a function **F** at ***2n+1*** evenly spaced points,

a, a+h, a+2h, ….., a+2nh (with ***h = b-a/2n***)

then we can estimated the integral $\int_{a}^{a+2nh} F(x)\,dx$ by

(h/3) [F(a) +*
4(F(a+h)+ 2F(a+2h)+ 4F(a+3h)+2F(a+4h)+ ... + 2F(a+(2n-2)h)+ 4F(a+(2n-1)h)+ F(a+2nh)]

That is the even position terms all have a coefficient of **4**, while the odd position terms have coefficient **2**, except for the first and the last, which have coefficient **1**.

Write a ML function **simpson** that takes starting and ending points ***a*** and ***b***, an a integer ***n*** (such that the evaluation is to use ***2n+1*** points as above), and a function **F** to integrate by Simpson's rule.

Code Snippet in #RollNo.Assign1.CS431.ml

```
-simpson(a:real,b:real,n:int,F) =
(*
    Write your code here and any other required functions
*);
-simpson(0.0,1.0,100, square);  (*need to define square *)
-simpson(0.0,1.0,100, cubeplus1); (*need to define cubeplus1 *)
```