C Language Programming: Homework #2
Assigned on 10/14/2014, Due on 10/21/2014

(For 一甲班)Let $\boldsymbol{a}$ be a positive real number, and let the sequence of real of real numbers $\boldsymbol{x_i}$ be given by

$$\boldsymbol{x_0 = 1,\ \ x_{i+1} = \frac{1}{2}\left(x_i + \frac{a}{x_i}\right) \quad for\ \ i = 0, 1, 2, \ldots}$$

It can be shown mathematically that $\boldsymbol{x_i} \to \sqrt{\boldsymbol{a}} \quad \boldsymbol{as} \quad \boldsymbol{i} \to \infty$

This algorithm is derived from the Newton-Raphson method in numerical analysis. Write a program that reads in the value of $\boldsymbol{a}$ interactively and uses this algorithm to compute the square root of $\boldsymbol{a}$. As you will see, the program is very efficient. (Nonetheless, it is not the algorithm used by the sqrt() function in the standard library.)

Declare x0 and x1 to be of type double, and initialize x1 to be 1. Inside a loop do the following:

```
x0 = x1;                /* save the current value of x1 */
x1 = 0.5 * (x1 + a / x1);    /*compute a new value of x1 */
```

The body of the loop should be executed as long as x0 is not equal to x1. Each time through the loop, print out the iteration count and the values of x1 (converging to the square root of a) and a – x1 * x1 (a check on accuracy)

(For 一乙班)The constant $\boldsymbol{e}$, which is the base of the natural logarithms, is given to 41 significant figures by

$$\mathbf{e = 2.71828\ 18284\ 59045\ 23536\ 02874\ 71352\ 66249\ 77572}$$

Define

$$\boldsymbol{x_n = \left(1 + \frac{1}{n}\right)^n \quad for\ \ n = 1, 2, \ldots}$$

It can be shown mathematically that $\boldsymbol{x_n} \to \boldsymbol{e} \quad \boldsymbol{as} \quad \boldsymbol{n} \to \infty$

Investigate how to calculate $\boldsymbol{e}$ to arbitrary precision using this algorithm. You will find that the algorithm is computationally ineffective. (See exercise 36, on page 195)

(For others 外系&重修)In addition to the algorithm given in the previous exercise, the value for $\boldsymbol{e}$ is also given by the infinite series

$$\boldsymbol{e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots}$$

The above algorithm is computationally effective. Use it to compute $\boldsymbol{e}$ to an arbitrary precision.