ECE3411 – Fall 2016
Lab 1a.

Introduction to C-Programming

Marten van Dijk, Chenglu Jin
Department of Electrical & Computer Engineering
University of Connecticut
Email: {marten.van_dijk, chenglu.jin}@uconn.edu

UCONN

Adopted from Lab1a slides "Introduction to C-Programming" by Marten van Dijk and Syed Kamran Haider, Fall 2015.

Prerequisites

- Eclipse development environment (with C Development Tools) installed
- Basic understanding of C Programming

Task 1: Approximate the value of π

- The value of π can be calculated by the following series expansion

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} + \cdots = \frac{\pi}{4} \quad \Rightarrow \quad \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}$$

- Task 1(a): Write a C program that takes a positive (≥ 0) integer n as input and prints the value of π computed up to the n^{th} term of the above series.
  - E.g. if $n = 3$ then the program computes $\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7}\right)$
- Task 1(b): Modify the program from Task1(a) such that it terminates only when the absolute value of the n^{th} term becomes less than 10^{-6}
  - Implement your own function to compute the absolute value of a double
- Task 1(c): Modify the program from Task1(b) such that it terminates only when the relative error in the π values from two consecutive iterations becomes less than 10^{-8}, i.e., when the absolute value of (pi-last_pi)/last_pi is less than 10^{-8}
  - The final output should be the π value from the most recent iteration, i.e. one with the higher value of n.

Task 2: Finding Prime Numbers

- A **prime number** (or a **prime**) is a natural number greater than 1 that has no positive divisors other than 1 and itself.
  - E.g. 2, 3, 5, 7, …
- Task 2: Write a C program which takes an integer as input from the user and prints all the prime numbers (separated by a comma) that are less than the entered number.
  - E.g. if the user inputs “10” then the program should print “2, 3, 5, 7”.