Program Design. Construct Algorithms using Problems Solving Tools
Sample Exercises: "el perrito"

File: SampleExercises-Sequential-v7.doc

IMPORTANT: Please do exercises pointed by ➡ and prepare for a quiz

Sequential Problems. These problems are for beginners. This set includes very simple problems which requires only of the sequence structure (no loops nor if-type logic control is required). The idea is to create an algorithm and express it by natural language or flowchart [FC] or pseudocode [PSC] helped by design tools, such as, decomposition with IPO diagram, cursory sketch [CS], and top down design [TDD]. You may be evaluated (quiz) with a similar problem, so be prepared.

Ex-1. Construct pseudocode (and flowchart) to salute the world in three different languages: 'Hello There World', 'Hola Mundo', 'Cio Mondo' on the screen. Please notice that this code doesn't have input nor processing. It's a simple program with just output. Now as second assignment write a **python** code and run it.

Ex-2. Construct a program to ask the user for someone's name and greets him/her with his/her name included. **Hint:** names can be stored as strings. E.g., "greeting" + name could be as "Hi Pepe."

Ex-3. Write a program that calculates the absolute difference between the sum and the product of two numbers and display the result.

Ex-4. Write a program (pseudocode and flowchart) that calculates the sum of powers of two numbers A and B as shown by the formula $R = A^B + B^A$ and display the result.

Ex-5. Differential Calculus. We are interested in comparing two formulas for computing the first derivative for the function $f(x) = x^3$ at x=3. In Differential Calculus, the derivative of a function f(x) at a point x is defined by the limit.

$$\frac{\boldsymbol{df(x)}}{\boldsymbol{dx}} = \boldsymbol{f'(x)} = \lim_{\boldsymbol{h\to 0}} \frac{\boldsymbol{f(x+h) - f(x)}}{\boldsymbol{h}}$$

The first derivative of $f(x) = x^3$ can be expressed as $f'(x) = 3x^2$, which evaluated at x=3, $f'(x = 3) = 3(3)^2 = 27$. Consider the true result.

Numerical Methods. Finite difference is often used as an approximation of the derivative, typically in numerical differentiation. Finite difference approximation for the first derivative:

$$\left.\frac{df(x)}{dx}\right|_x = \frac{f(x+h)-f(x)}{h} + \vartheta(h), \text{ where } h = (x+h) - x$$

For the initial computation with the finite difference formulation use h=0.001. You can reduce the error by decreasing h.

Ex-6. Write a program (pseudocode and flowchart) to calculate the surface area and volume of a cylinder and display the results (Recall $A = 2\pi rh + 2\pi r^3$ and $V = \pi r^2 h$ where pi (use pi instead of π) is approximately equal to 3.1416, **r** is the radius and **h** is the height). **NOTE:** Flowgorithm does not have a predefined parameter pi. While python π is simple math.pi. In C language, we must define a constant as #define pi = 3.1415 ... to whatever depth you need (Alternatively, #Include <math.h> header file and use the symbol M_PI, not all compiler implements this one though). See appendix for implementations in VBA.

Ex-7. Gone with wind

Ex-8. Write a program (pseudocode/flowchart) to compute the standard deviation of the following set of eight numbers: x_1, x_2, x_3, x_4, x_5, x_6, x_7, and x_8, and then displays the result. The standard deviation **σ**, sigma, formula is:

$$\boldsymbol{\sigma} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{N}}(\boldsymbol{x_i} - \boldsymbol{\mu})^2}$$

where x_i is each individual value, μ is the set average and **N** is the number of data values. Later we will learn there are others ways to solve this problem more efficiently, e.g., using array variable and loops

Ex-9. Write a program (pseudocode/flowchart) that converts Fahrenheit degrees into Celsius and Rankine degrees. Use the formulas: C= (5/9)*F-32 and R = F + 459.67

Ex-10. Gone with the wind

Ex-11. Construct a program to evaluate $\ln(x^2+1)$ and its derivative, $\frac{d\ln(x^2+1)}{dx}$ at $x_1 = 2$ and $x_2 = 3$. Note that $\frac{d\ln(x^2+1)}{dx} = \frac{2x}{(x^2+1)}$

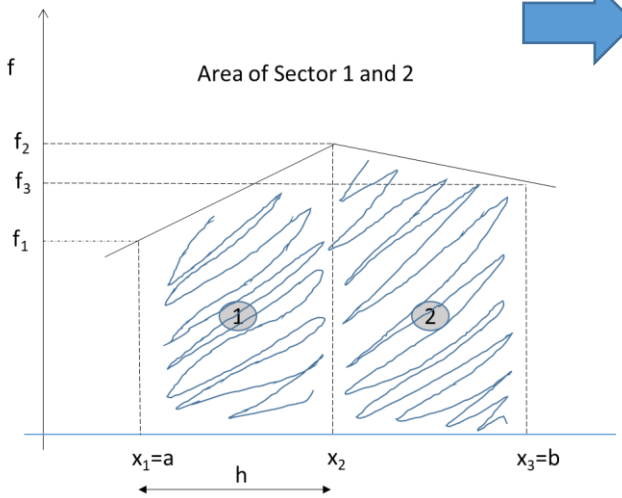
Ex-12. Construct an algorithm to evaluate the function $f(x) = e^x$ at x=3.

Ex-13. Write a program to evaluate the derivative of f(x) at x=5, where $f(x) = 3x^2 + 2x + 1$

Ex-14. Gone on vacation

Ex-15. Gone on vacation

EX-16. Construct a program (pseudocode and flowchart with flowgorithm) to report the remainder of N/D division, where N and **D** are integers (Hint: use the mod (or modulus or modulo) operator or function). Later improve the code to test if N is an odd number [implement a selection structure].

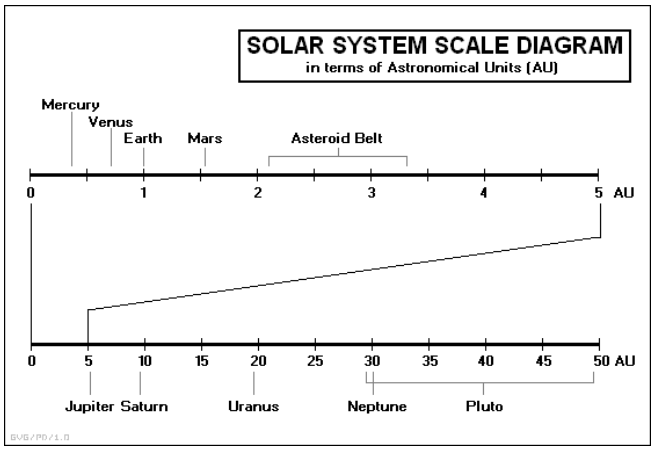


Ex-17. Area under the curve. Shown is the formula to compute the total composed area of trapezoidal-1 plus trapezoidal-2:

$$Total\ Area = \frac{h}{2}(f_1 + 2f_2 + f_3)$$

Design a program (flowgorithm flowchart /pseudocode) to compute the composed area under the curve given n [number of sectors], f_1, f_2, and f_3 [functions evaluated at x_1, x_2, x_3] where h [step size] is a constant given by $h = \frac{(b-a)}{n}$; b (right end point); a (left end point); and n=2. Test your program with f_1=2, f_2=3, f_3=2.5; x_1=1, x_2=3, x_3=5, and n=2.

Ex-18. Write a program (pseudocode) to convert the distance of the nine major planets from AU to km. Background: The illustration below depicts the true relative distances from the Sun of each planet as well as asteroids in the Asteroid belt, measured in astronomical units or (AU) which is equal to the length of the semi-major axis of the Earth's elliptical orbit or 149,597,870.69 km. Notice that the Earth is positioned at 1 AU and the farthest major planet Neptune is positioned at 30 AU. Also notice that Pluto's distance from the Sun varies between 30 and 50 AU, with its orbit sometimes crossing that of Neptune



http://people.bu.edu/sscruggs/majorplanetshome.html)

Ex-19. Write a program (pseudocode/flowchart with flowgorithm) that calculates the sum of powers of two numbers A and B as $R = A^B + B^A$ and display the result. The number B must be half of three times A.

Ex-20. A biotechnologist is researching a newly-discovered species of bacteria. At time t = 0 hours, he puts one hundred bacteria (A_0=100) into what he has determined to be a favorable growth medium. He wants to compute how many more bacteria will be after six hours. Assuming exponential growth, the growth constant k for the bacteria is 0.25/hr. The amount of bacteria **A** after an elapsed time t (starting at t_o=0), is $A = A_0 e^{kt}$ (See http://www.biology-pages.info/P/Populations.html). Design a "flowgorithm" flowchart and pseudocode.

Ex-21. Chemists define the acidity or alkalinity of a substance according to the formula pH = –log[H+] where [H+] is the hydrogen ion concentration, measured in moles per liter and log is the natural logarithm. Solutions with a pH value of less than 7 are acidic; solutions with a pH value of greater than 7 are basic; solutions with a pH of 7 (such as pure water) are neutral. Suppose that you test your program with apple juice with a hydrogen ion concentration of [H+] = 0.0003. Find the pH for the apple juice.

Ex-22. Construct a program (pseudocode and flowgorithm flowchart) to compute the area (A) and perimeter (P) of a regular pentagon.

Ex-23. Construct (IPO diagram, pseudocode, flowchart) for three different problems aiming to compute the equation of a line expressed as y = mx + b, given different input:

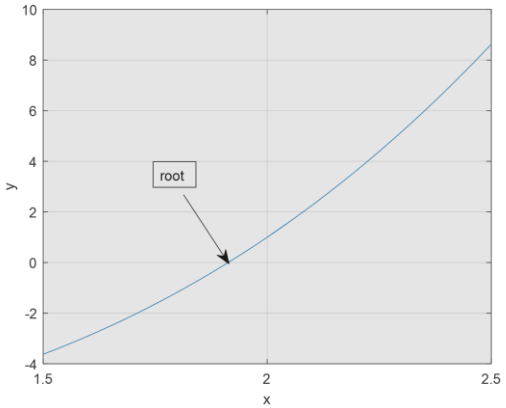
(A) the coordinates of two points (x_1,y_1), (x_2,y_2); where m is the slope and b the y intercept.
(B) the coordinates of one point (x_1,y_1), and b (the y-intercept).
(C) The slope "m" and the y-intercept, "b."

Remark: Above there are three different programs.

Ex-24. Construct a program to switch the value stored in A to B, and the value stored in B to A. E.g., if A=1 and B=3, the new values after switching are A=3 and B=1. Trace your code solution. Test your code with another set of values, such as A=7, B=5. Your code should work for any values.

Ex-25. Write a pseudocode (then python; yes python) to transfer the value of A to C, the value of B to A, and the value of C to B.

Ex-26. Write a program to get **one step** closer to the root of the function $f(x) = x^3 - 7$ if we utilize as the starting point $x_0 = 2.0$ in the Newton's method. Newton's state that an improved estimate to the root is $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$, where x_i is the starting estimate of the root (2.0), x_{i+1} the new estimate, $f(x_i)$ the function $x^3 - 7$ and $f'(x_i)$ the derivative $3x^2$, both evaluated at x_i. A graph of the function:



For the first iteration, indices become:

$$x_1 = x_o - \frac{f(x_o)}{f'(x_o)}$$

Level-0-Algorithm

1. Input xo
2. Compute fo, fpo
3. Compute x1
4. Output x1.

Solution:

input	processing	output

x_0	f_0 fp_0 $x_1 = x_0 - f_0/fp_0$	x_1

The graph was constructed with the matlab code:

```
% PlotDemo101.m
clc, clear, clf
f=@(x) x^3-7
fplot(f,[1.5,2.5])
grid on
xlabel('x'); ylabel('y');
```

One problem in AC or DC currents (https://opentextbc.ca/physicstestbook2/chapter/alternating-current-versus-direct-current/

Discontinued or Gone with the quake:

Ex-5. Construct a program (pseudocode or flowchart) to calculate the area of a circle and display the result. (Recall: A= π r2; where pi [use pi instead of π] is approximately equal to 3.1416).
Ex-7. Write a program (pseudocode/flowchart) that computes the average score of five quizzes, and then display the result.
Ex-10. Create pseudocode/flowchart to compute the volume of a sphere. Use the formula: V= (4/3)* π r^3 where π is equal to 3.1416 approximately and r is the radius. Display the result
Ex-27. Write a program to get **one step** closer to the root of the function $f(x) = x^3 - 7$ if we utilize as the starting points $a = 1.5$ and $b = 2.0$, values bracketing a root in the Bisection's method. Bisection states that an improved estimate to the root is obtained by $c = \frac{a+b}{2}$, where a and b are the extremes of the root bracketing interval, c the new improved estimate, $f(a)$ the function evaluate at $x = a$ and $f(b)$ the function evaluate at $x = b$. A graph of the function: