MA348 Warmup On Unix, Fortran and Gnuplot

This exercise involves most of the skills/tools needed in computation: programming, Unix, editing, compiling, executing, plotting ... Pay attention to the results of your actions so you'll be learning. Don't worry if you don't understand everything yet, we'll study all these aspects in detail. Don't get frustrated, it gets easier fast!

A Fortran code is given (see page 3) that evaluates the function: $F(x) = a-bx^3$ at N +1 equidistant points in [0; 1], and outputs the pairs (x; F(x)). The user is asked to enter values for a, b, and N.

Do the following:

1. Open a terminal and type:

mkdir Lab1	to create a directory named Lab1
ls	to see that the dir "Lab1/" has been created
cd Lab1	to go to the directory Lab1

2. Type the code in a file named "lab1.f" inside the dir Lab1. BE CAREFUL, no misprints allowed! Be careful all indented lines start on the 7^{th} column.
You can use vi or any other editor.
If you use vi then type: vi lab1.f
then press: i to enter insert mode;
Look at the HOW-TO-ESSENTIALS (everything you need in a page) handout for vi commands and try to learn them. As you type, save your work frequently: ESC (to command mode) and :w (write) (so if anything goes wrong you won't lose everything !) When you are typing the code, press ESC (to make sure you are out of insert mode), save it :wq.

3. Compile it: f77 -o go lab1.f go will be the name of your executable
Type ls and see that an executable go has been created.
Remark : if you do not mention any specific name for your executable the default name is a.out, you can try.

4. Run it: ./go

Enter: 2.0 1.0 5 (which the program reads as: a b N)
You should get: Thanks, will run with: a= 2.0000000 , b= 1.0000000 , N= 5

x	F(x)
0.0000000	2.0000000
0.20000000	1.9920000
0.40000001	1.9360000
0.60000002	1.7839999
0.80000001	1.4880000
1.0000000	1.0000000

All Done, BYE !

5. Try various choices: a = -1.0, 0.0, b = 0.1, 5.0, N = 20

6. The program also writes the values in a file named "fort.7" (that's what the statement: write(7,*) does), appropriately formatted for plotting with a graphics tool, called gnuplot. All our machines have it installed (it comes with Linux). Read about gnuplot in the HOW-TO-ESSENTIALS (everything you need in a page) handout.

Note: Lines with # in first column are ignored by gnuplot, that's why we inserted # at the start of the first line written to "fort.7".
Use gnuplot to display the graph in each case: Move mouse to another terminal, make sure you are in Lab1/ (cd /Lab1), type:
gnuplot plot "fort.7" with points
q (to close the plot)
plot "fort.7" with linespoints
Check it by plotting the curve itself (say, for a = 1; b = 0:1):
*plot 1-0.1*x**3 with lines*
For easy comparison, plot your values and this curve on the same plot:
*plot "fort.7" with points, 1-0.1*x**3 with lines*
Play with N, to see how many points you need to make them look like the same curve (the points on the line).
(Note: Ctrl-P in gnuplot backtracks to previous commands, try it to save re-typing !)

7. Now sit back and think about what you have learnt.
To do computing, you need to:
- create a code (how can you do this?)
- compile the code (how?)
- run the code (how?)
- look at the results and possibly plot them (how?)

8. Start getting into some good habits:
Always clean up your files when your lab is done! In particular,
- delete any junk files you may have created;
- delete your executable (executables are large files, wasting disk, and can always be regenerated by recompiling);
- move any file pertaining to Lab1 into the Lab1 directory.
The only file you really need to keep is your code (lab1.f)
10. When you are finished, exit from the browser, vi, gnuplot, ... and finally log out.

9. Print a few of you plots giving the parameters you used to get them along with your code and submit it as part of HW1.

```
c234567890123456789012345678901234567890123456789012345678901234567890123456789012
!------------------------------------------------------------------------!
      PROGRAM Lab1
! Purpose:
! Evaluate F(x) = a - b*(x**3) at N+1 equidistant points in [0,1],
! and output the pairs x , F(x) on the screen and in file fort.7
! for plotting.
!________________________________________________________________________
       IMPLICIT NONE
! Declare the variables used in this program
       Double precision a,b
       integer N,i
       Double precision Dx,xi,Fi

! Get input from user:
      write(*,*),'Please enter values for a, b, N :'
      read(*,*), a, b, N
      write(*,*),'Thanks, will run with:'
      write(*,*),' a=',a,' , b=',b,' , N=',N
      write(*,*)'     '
      write(*,*),' x F(x)'
      write(7,*)'# Output from Lab1 with: a=',a,' b=',b,' N=',N
c Compute F(x) = a - b*(x**3) and print x F(x):
      Dx = 1.0 / N
      DO i = 0, N
       xi = i * Dx
       Fi = a - b * xi**3
      write(*,*), xi, Fi
      write(7,*) xi, Fi
      ENDDO
! Exit:
      write(*,*),'All Done, BYE !'

      END
```