

Simple Matlab Exercises for Matlab Grader

Don R. Baker

Please note: You will have to read about the various functions used in these exercises in the Matlab help files or on the web.

Exercise 1 – simple input and output to screen

Almost any program you write needs to have data for a calculation and then needs to output the data. This assignment gives you practice with simple input and output. What you need to do is create a program that asks you to put in the slope, m , and intercept, b , of a line and then display those numbers to you using `disp` and `fprintf`.

Here is the pseudocode for the problem:

Use the function "input" to input the slope, include the words "The value of the slope is" in the statement -- use a value of 5 for the slope

Use the function "input" to input the intercept, t , include the words "The value of the intercept is" -- use a value of 3 for the intercept

Use the function "disp" to display the slope

Use the function "fprintf" to print out "The intercept of the line is" and then print out the intercept on the same line. Note that you will need the format instruction "%f"

Note that the number of empty lines in the Learner Template DOES NOT correspond to the lines of code that need to be written.

Reference solution

```
% m is the slope -- use a value of 5
% b is the intercept -- use a value of 3
m = input('The value of the slope is ')
b = input('The value of the intercept is ')
disp(m)
fprintf('The intercept of the line is %f\n',b)
```

Assessment

Test 1: Is the variable m entered correctly? (m = Reference Solution?)

Test 2: Is the variable b entered correctly? (b = Reference Solution?)

Test 3: Was the correct function used? (`disp` present?)

Learner template

Nothing provided

Exercise 2—Loops: for and while statements

The power of a computer programming language is its ability to perform repetitive tasks quickly, and an example might be trying to find the conditions at which the ΔG of a reaction equals 0 (and therefore products and reactants are both stable) by changing the temperature of the reaction at a fixed pressure and composition.

To construct a loop you need to provide a starting point and a stopping point. For example, let's construct a loop that counts from 0 to 10 by adding 1. We can start the script by writing `count = 0`, but how do we end the loop stop at 10? There are two commands that are commonly used "for" and "while". Each of these command precedes one or more lines of executable code and the statements are usually terminated with the command "end".

The "for" command repeats a loop for a number of steps that are specified, as an example: the following code will display the value of `i` as the loop repeats 10 times:

```
for i=1:10
    disp(i)
end %end if
```

Note: you can test this in the command window by writing: `for i=1:10;disp(i);end`

The for command does not have to count by integers. In the previous code replace "for i=1:10" with "i=1:0.25:10" and watch what happens.

The "while" command is a little different and requires a certain condition to met for an action to take place. In the example above of the "for" loop it only counted to 10, to write an equivalent loop with a while command is more complicated, but those complications allow much more flexibility than a "for" command can offer:

```
i = 1
while i <=10
    disp(i)
    i = i + 1;
end % end while
```

This can be run in the command window as: `i=1; while i<=10;disp(i);i=i+1;end` Note the value of `i` after this command has been executed by typing "i".

Write a script based upon the following pseudocode

start with a value of `x = 1000`

Using a "for" loop calculate and show the value `x` for 10 loops where `x` increases by 4% in each loop (`x = x*1.04`)

`fprintf` the value of `x` after this loop is completed

set the variable "total" equal to `x` (for testing purposes)

`r`

reset count to 0

Now using a "while" loop to determine how many steps it would take for `x` to be below 1000 if `x` is reduced by 2% each loop (`x=0.98x`). Use the variable "count" to count the steps

`fprintf` the number of steps

Note that the number of empty lines in the Learner Template DOES NOT correspond to the lines of code that need to be written.

Reference solution

```
x=1000;
```

```
for count=1:10
```

```
x=x*1.04;
```

```
fprintf('x = %f\n',x);
```

```
end
```

```

total = x;
count = 0;
while x > 1000
x = 0.98*x;
fprintf('x = %f\n',x);
count = count+1;
end
fprintf('count = %f\n',count)

```

Assessment

Test 1: Was the final “count” correct? (count = Reference Solution?)
 Test 2: Was the value of “total” correct? (total = Reference Solution?)
 Test 3: Were the correct commands used? (for, while fprintf present?)

Learner template

```

x=1000;
for

end

total = x;

while
x = 0.98*x;

end
fprintf('count = %f\n',count)

```

Exercise 3 – The conditional statement “if”

The “if” command typically compares to variables and determines if they are equal to each other (or true) or if one is greater than (or greater than or equal to) the other (in this case when the count equals 10). When the “if” statement is true then an action specified by the programmer occurs—such as:

```

if hungry == true (note use of double equals sign, can also use “>”, “>=”, “<”, “<=”)
    eat
end    %(note the use of the “end” statement)

```

An auxiliary command to “if” is “else” and it executes if the if command is not fulfilled. For example:

```

if hungry == true
    eat
else

```

```
study geochemistry
end %end if
```

So now try to code the following problem from the pseudocode
generate a random number (use the function “rand”)
multiply that number by 1000
if the number is less than 500 fprintf the value together the words “less than 500” (or something similar)
If the number is greater than 500 fprintf the value together the words “greater than 500” (or something similar).
Note that the number of empty lines in the Learner Template DOES NOT correspond to the lines of code that need to be written.

Reference solution

```
x = 1000* rand;
if x < 500
fprintf('The value of x is less than 500: %f\n',x)
else
fprintf('The value of x is greater than 500: %f\n',x)
end
```

Assessment

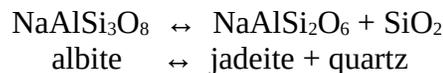
Test 1: Are all the keywords and functions present? (rand, if, else, fprintf present?)

Learner template

```
x=1000*rand
```

Exercise 4 – Calculate the equilibrium temperature of a reaction and plot the results

Calculate and plot the equilibrium temperature of the following reaction at pressures from 100 MPa (1 000 bars) to 1000 MPa (10 000 bars) in 100 MPa (1 000 bar) steps by finding where the ΔG of the reaction (abbreviated rxn) is 0 (or close to 0, within 50 J):



The equation to solve is:

$$\Delta G_{\text{rxn}} = \Delta H_{\text{rxn}} - T\Delta S_{\text{rxn}} + (P-1) \Delta V_{\text{rxn}} = 0 \text{ at equilibrium}$$

The values (which you may assume remain constant) are:

$$\begin{array}{l} \Delta H_{\text{rxn}} = -10040 \text{ J} \\ \Delta S_{\text{rxn}} = -43.39 \text{ J/K} \\ \Delta V_{\text{rxn}} = -1.7342 \text{ J/bar} \quad (1 \text{ bar} = 0.1 \text{ MPa}) \end{array}$$

T is always in K (degrees Kelvin)

In order to write the script to solve this problem you will need to use what you have learned in Exercises 1 to 3; you need to input the data, you need to use a “for loop” to change the pressures, and you need a “while loop” to change the temperatures in order to find where $\Delta G_{\text{rxn}} = 0$. Writing the program will require you to create two arrays of answers, TK(i) and pressure(i), where i ranges from 1 to 10. And also, you need to learn how to use the plot command to print out the line of equilibrium where the products and reactants are stable by plotting the results of your calculations with T(K) on the x-axis and P (in bars or MPa) on the y-axis.

Note that the number of empty lines in the Learner Template DOES NOT correspond to the lines of code that need to be written.

Reference solution

```
% place input values after this line
```

```
DelH = -10040; %J
```

```
DelS = -43.39; %J/K
```

```
DelV = -1.7342; % J/bar
```

```
%Now write a for loop to step through pressure from 1000 to 10 000 bars
```

```
%in 1000 bar steps
```

```
i = 1;
```

```
for P=1000:1000:10000
```

```
pressure(i) = P;
```

```
DelG = 10000;
```

```
TK(i) = 200;
```

```
%Now write a while loop to calculate DelG until its absolute value is less than 50 J
```

```
while abs(DelG) > 50
```

```
TK(i) = TK(i)+1;
```

```
DelG = DelH - TK(i)*DelS + (pressure(i) - 1)*DelV;
```

```
end % end while
```

```
fprintf('T(K) = %f, P(bars) = %f\n',TK(i),pressure(i))
```

```
i= i+1;
```

```
end % end for
```

```
plot(TK,pressure)
```

```
Tfinal = TK(10);
```

Assessment

Test 1: Are the proper commands used? (while, plot, abs, for present?)

Test2: Is the final temperature at 1 000 MPa (10 000 bar) correct? (Tfinal = Reference Solution?)

Test 3: Are the products stable at high or low pressure? (ProductsStableLowPressure = Reference Solution?)

Learner template

```
%AlbiteJadeite_Ex4.m
```

```
% place input values of the thermodynamic variables
```

```

%Now write a for loop to step through pressure from 1000 to 10 000 bars
%in 1000 bar steps
i = 1; % a counter is needed to keep track of the solution (TK) at each pressure
for P = % This command needs to be completed
pressure(i) = P;
DelG = 10000; % at each pressure DelG must be reset to a high value so the while loop calculates the
new, correct, temperature
TK(i) = 200; % at each pressure the temperature must be reset so the while loop calculates the new,
correct, temperature
%Now write a while loop, within the for loop, to calculate DelG by increasing the temperature by 1
degree each time until its absolute value is less than 50 J
while abs(DelG) > 50

end % end while – the equilibrium temperature has been found
fprintf('T(K) = %f, P(bars) = %f\n',TK(i),pressure(i)) % note the variable hint
i= i+1; % note that i is incremented to keep track of the results at each pressure
end % end for
%now plot the results that are in arrays TK and pressure

Tfinal = % Tfinal needs to be set equal to the final equilibrium temperature for test 2

```