

Department of Chemical Engineering
ChE-1800: Approaches to Chemical Engineering Problem Solving
MATLAB Tutorial II

Quality Program and Script m-files

(last updated 4/04/06 by GGB)

Objectives:

These tutorials are designed to show the introductory elements for any of the topics discussed. In almost all cases there are other ways to accomplish the same objective, or higher level features that can be added to the commands below. Before working on this tutorial you should understand the concepts described in H-2 (flowchart diagrams).

Any text below appearing after the double prompt (>>) can be entered in the Command Window directly or in an m-file.

The following topics are covered in this tutorial;

Introduction: General Rules for Algorithms and Codes**Creating and Executing a Script m_file****Matlab Prompted Input _ The "input" Command****Menu Boxes for User Choices****Display Text or Variable Values on the Screen _ The "disp" Command****Decision Making: if_else syntax****While loops****Solved Problems (guided tour)****Proposed Problems**

Introduction: General rules for algorithms and codes:

It is important that you know how to build flowchart diagrams. Before you write a code you should build your flowchart diagram. Please refer to H-2 to learn more about it. Once you have built your flowchart diagram you are ready to write your code. Below are described some basic tips that will help you write a quality program:

1.1 Start with a simple program. Do not add cosmetic features up front.

As a general rule you should develop your algorithms as simple as possible. Include only the fundamental steps that you need to solve the problem. That means that you will build first a simple program that is able to compute the calculations required.

Once your simple program works, you can make cosmetic changes to make your program more automatic and user friendly.

1.2 Document your code. You should add as many comments as possible to your program so anybody (including yourself days later) can understand your code. Typical comments include (but are not limited to):

- ✓ Description of the function, program or subroutine
- ✓ Description of input variables and output variables
- ✓ Author name
- ✓ Dates when the program was built and last modified

- ✓ Include comments to the major calculations make by the program
- ✓ Explain the major loops

Script m-files

While Matlab commands are relatively easy to type and execute, it can be tedious when a long list of commands needs to be executed several times, may need to be edited, or stored for later. Saving a Matlab workspace only saves the values of the variables in memory, not any of the commands. Matlab gives you a way to handle this by allowing a list of commands to be stored in a simple text file. You can then open the text file and Matlab will execute the commands as if they were typed at the Matlab prompt. These text files are called **script files** or **m_files**. The term *m_file* comes from the fact that the script files must have the '.m' extension, **filename.m**. Summarizing, *script files* are user-created files with a sequence of MATLAB commands in them.

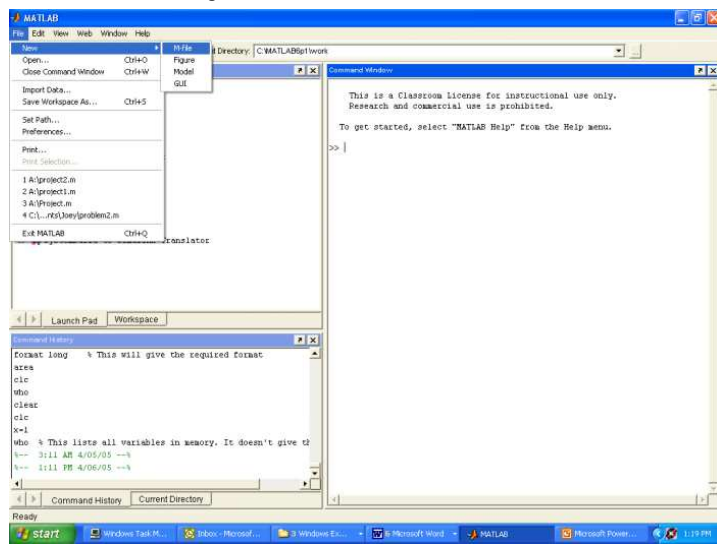
The commands in the *m_file* are executed from the workspace. Any variables in the Command Window are available to the *m_file* and any variables defined in the *m_file* will be stored in the Command Window memory upon execution.

Creating and Executing a Script m_file

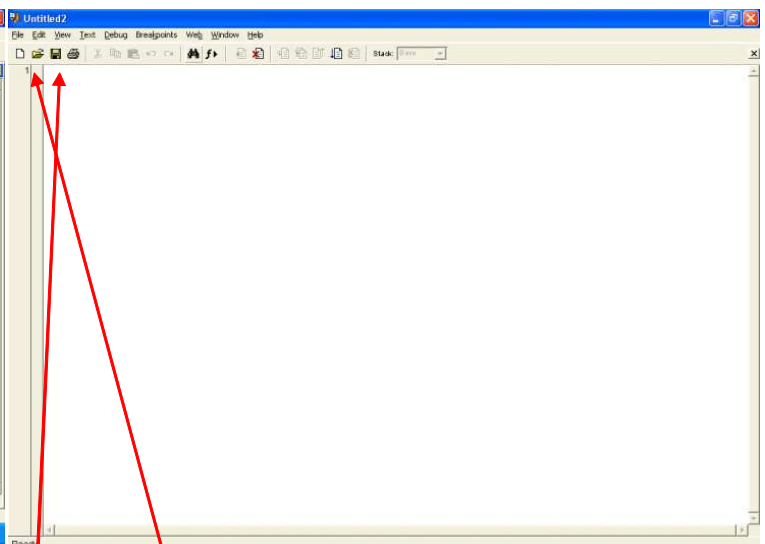
Creating a script m_file:

- ! Click on FILE/NEW/M_FILE. This will open the Matlab Editor/Debugger. This is not another Command Window. Matlab commands cannot be executed from here. Alternately, the NEW FILE icon on the toolbox can be clicked. See the figures below

Step 1: Opening the script file window



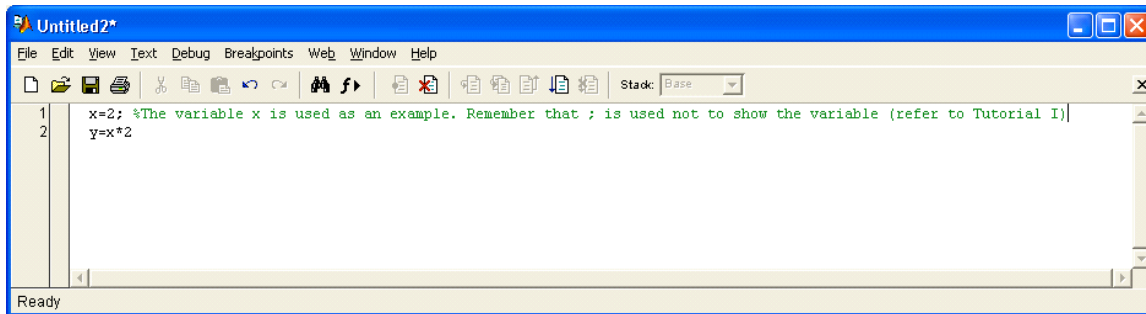
Step 2: This is how the script file window looks like



You type your commands in this area. Use different lines for different commands or instructions in the code

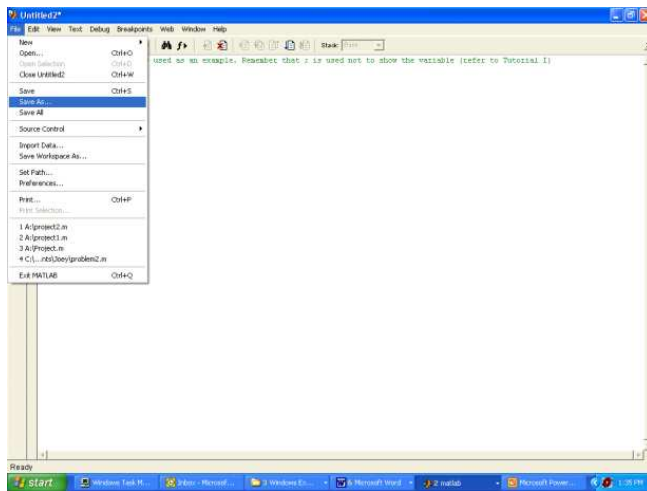
Line #, this is important when debugging the code as it will tell you which line has a problem

- ! Enter the Matlab commands you want to execute. Remember the use of the semicolon to suppress output if desired. Also be sure to use comments (the % symbol) spaces, and indents to make your code more readable. See an example below

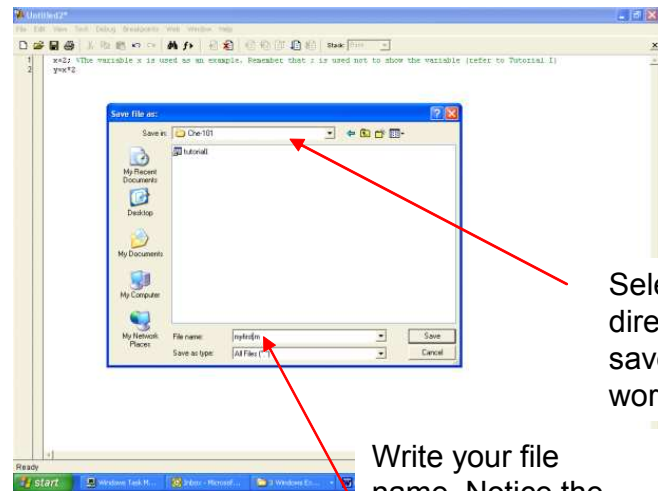


- ! Save the file by clicking on FILE/SAVE AS in the text editor. Make sure you are in the directory you want. The 3.5" floppy disk is the A drive and the zip disk is the E drive. Give it a name **xxx.m**. IT MUST HAVE A '.M' EXTENSION OR IT CAN'T LATER BE EXECUTED. See figure below

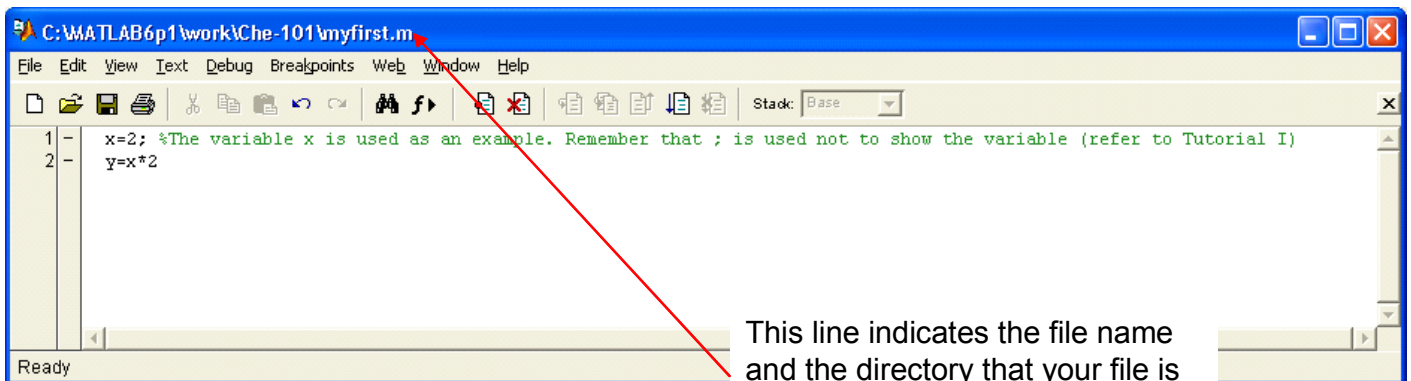
Step 1: Select "Save As"



Step 2: File name and directory



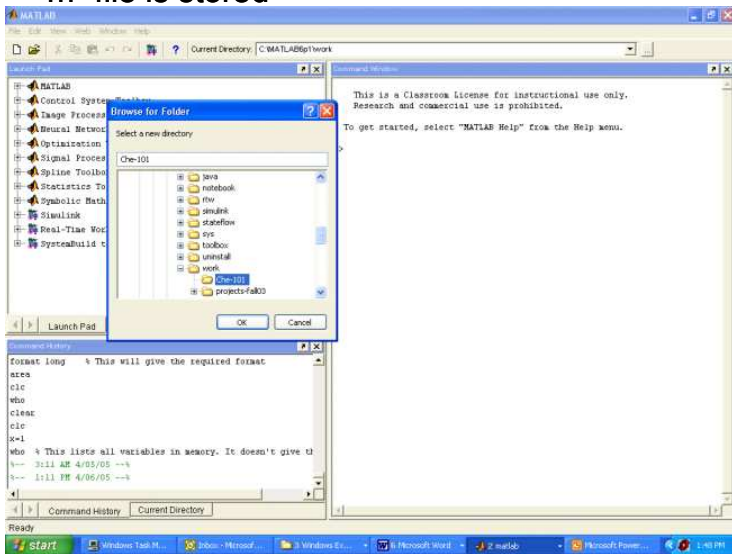
Step 3: File is already saved



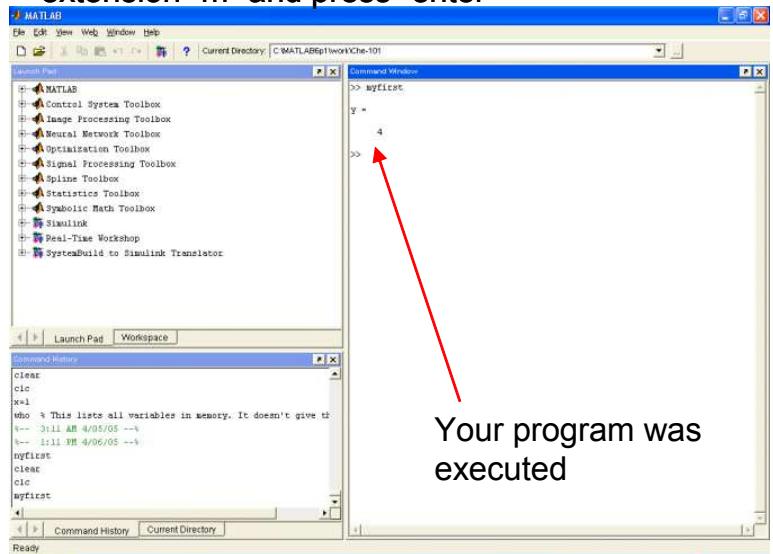
Executing a script m_file:

- ! Return to the Matlab Command Window.
- ! Change to the directory you where the m_file is located. For the 3.5" drive this is done by entering the command `>> cd a:`. For the zip drive this is done by `>> cd e:`.
- ! To execute the m_file type the filename **xxx**, without the .m extension at the Matlab prompt. Matlab will first look to see if **xxx** is a variable, then it looks to see if it is a built in Matlab command, then it looks for a script m_file. For example 'aaa' can be both a variable and there can be an m_file **aaa.m**. If so, Matlab will just report the value of the variable.

Step 1: Change to the directory where your “m” file is stored



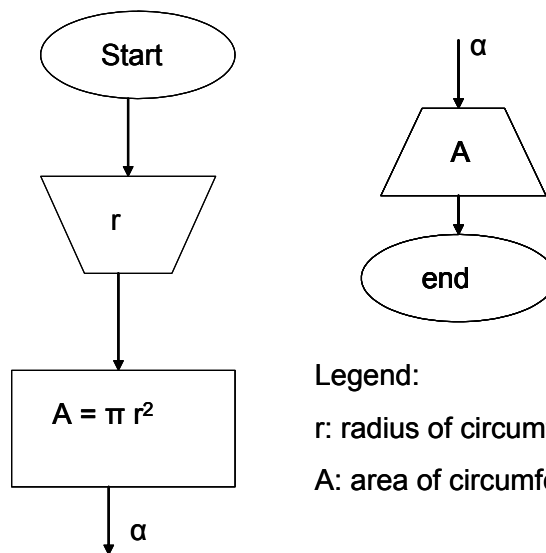
Step 2: Type your file name without the extension “m” and press “enter”



If you want to see the commands as they are executed click on **OPTIONS / TURN ECHO ON** or type `echo on` at the Matlab prompt. This is necessary if you want to see all the comments. Be careful if you have an m_file with loops that may execute many times.

Solved Problem 1:

Write an algorithm in Matlab to calculate the area of a circumference. The flowchart diagram was developed in class (see H-2).



Type your code as shown below (create a new m file window):

```

1 % This program calculates the Area of a Circumference given the radius
2 % Developed by Gerardine Botte
3 % Created on: 04/06/05
4 % Last modified on: 04/06/05
5 % Che-101, Spring 05
6 %
7 % Description of variables:
8 % The user will be prompt to input the value of the radius of the circumference (r)
9 % The area of the circumference is given in the variable "A"
10 -----
11 clear
12 A=pi*r^2 % r is the radius.
    
```

Notice all the comments in the code

Go to the command window and type “help area_circumference” What did you get?

```

>> help area_circumference

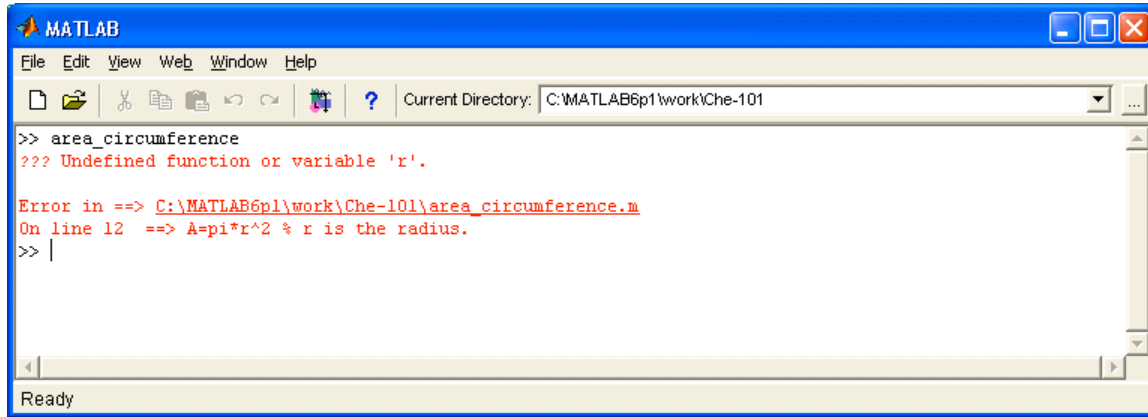
This program calculates the Area of a Circumference given the radius
Developed by Gerardine Botte
Created on: 04/06/05
Last modified on: 04/06/05
Che-101, Spring 05

Description of variables:
The user will be prompt to input the value of the radius of the circumference (r)
The area of the circumference is given in the variable "A"
-----

>> |
    
```

Every comment line that you wrote before the first command will show up when you try to use the online help for your code. This is a great help for the user, and this is one of the reasons why you should get used to do that for every code you write in Matlab

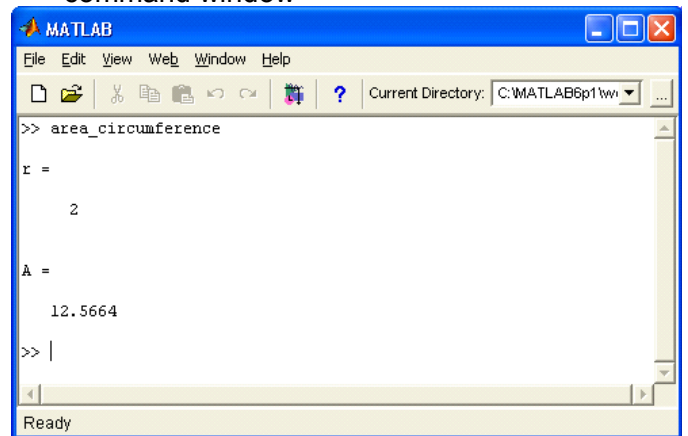
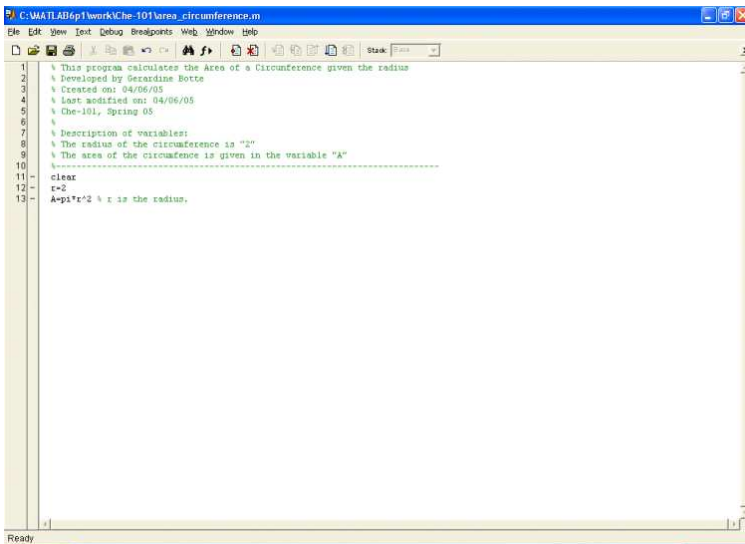
Run your program in the command window. Does it work? What do you think happened?



How would you solve this problem?

Step 1: Modify the m file and save it

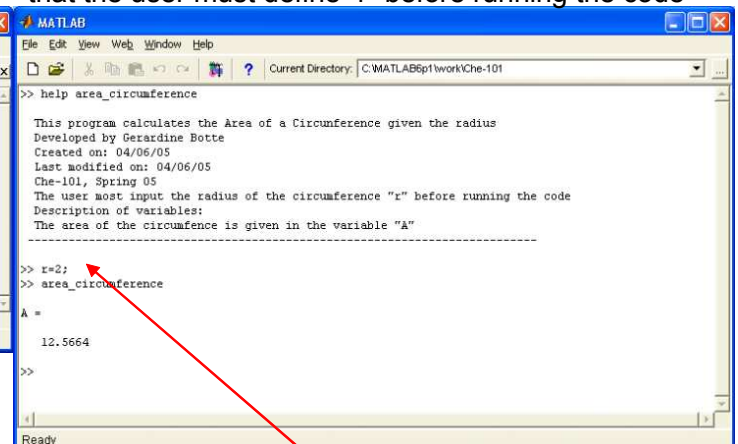
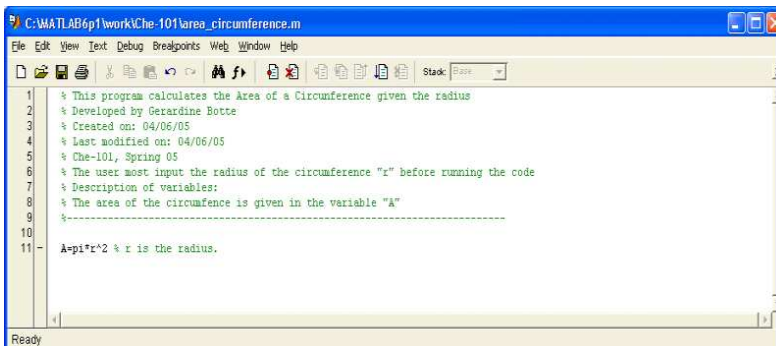
Step 2: Execute the program in the command window



Other Experiments: Modify your code as shown and run the program now:

Step 1: Modify your script file and save it

Step 2: Check on the help "online" Notice that it says that the user must define "r" before running the code



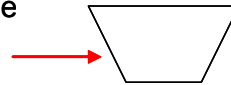
r is defined in the command window

Change the value of “r” and see how the area of the circumference is calculated.

CONCLUSION: to make your programs useful it is NECCESSARY that you learn how to program to allow the user INPUT values to the variables while your program is running. Please refer to the next sections to learn how to do that.

Input Options

This Matlab command is the equivalent to the following unit in a flowchart diagram, see H-2



Matlab offers a number of methods of entering or display data, both on the screen or for submission. Effective presentation of your results is an important step in any problem or project. This part of the tutorial will cover input options other than the default settings in Matlab.

Matlab Prompted Input _ The "input" Command

You can have MATLAB prompt you for input at any time by using the **input** command. This command will display a text string, specified by you, and then wait for input. Be sure to tell the user how to enter the data, especially if it is to be entered as an array.

Example: Modify the program to calculate the area of the circumference so that the user is prompted to input the radius. Notice that everything between (' ') is what they user will see in the screen. Therefore, you should include all the information needed in this area.

OPTION 1

Step 1: Modify the input file accordingly

Step 2: Run the program. The user is prompted to input the radius

```

1 % This program calculates the Area of a Circumference given the radius
2 % Developed by Gerardine Botte
3 % Created on: 04/06/05
4 % Last modified on: 04/06/05
5 % Che-101, Spring 05
6 % The user will be prompt to input the radius of the circumference
7 % Description of variables:
8 % The area of the circumference is given in the variable "A"
9 %-----
10 r = input('radius of the circumference in cm r = '); % r is given by the user
11
12 A=pi*r^2 % A is the area.
    
```

```

>> area_circum_input1
radius of the circumference in cm r = 2

r =

     2

A =

 12.5664

>> |
    
```

OPTION 2

Step 1: Modify the input file accordingly

Step 2: Notice the difference after running the program

```

1 % This program calculates the Area of a Circumference given the radius
2 % Developed by Gerardine Botte
3 % Created on: 04/06/05
4 % Last modified on: 04/06/05
5 % Che-101, Spring 05
6 % The user will be prompt to input the radius of the circumference
7 % Description of variables:
8 % The area of the circumference is given in the variable "A"
9 %-----
10 r = input('radius of the circumference in cm r = '); % r is given by the user
11
12 A=pi*r^2 % A is the area.
    
```

```

>> area_circum_input1
radius of the circumference in cm r = 2

A =

 12.5664

>> |
    
```

With this option the radius is not reprinted on the screen

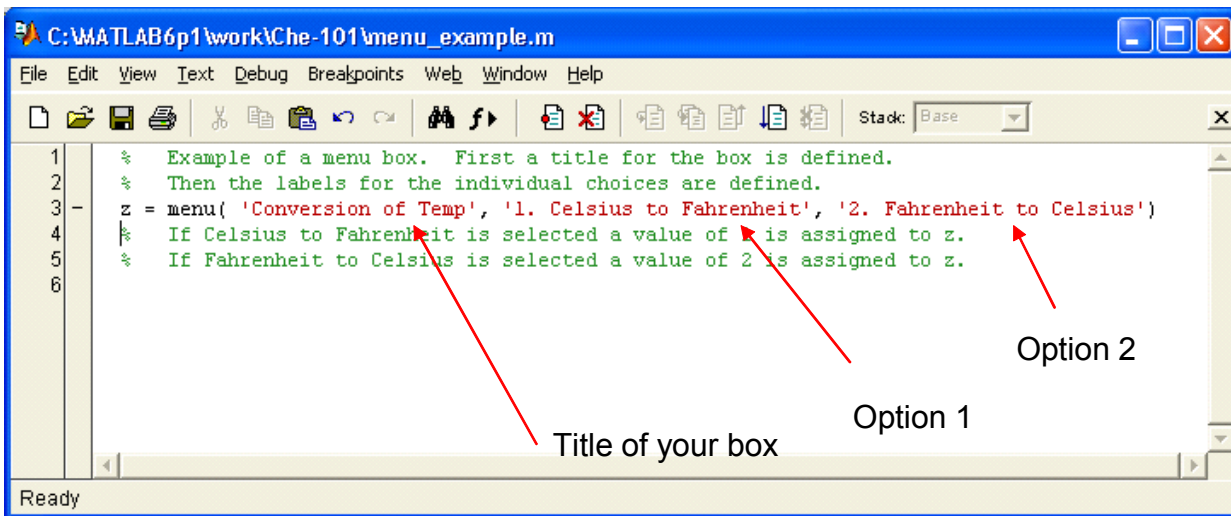
A semicolon at the end of the command will cause the on-screen display of the inputted values to be suppressed as shown above.

Menu Boxes for User Choices

It is easy to create a menu box that allows the user to select between different options. The menu box will assign a numerical value to a defined variable. The user would then use an if / else / elseif block to execute the appropriate commands.

The creation of a menu box is demonstrated in the following section of MATLAB code. Reproduce the example given below

Step 1: Create the menu script



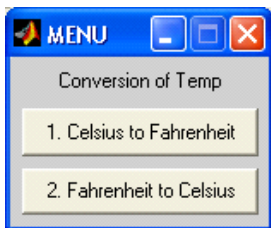
```
C:\MATLAB6p1\work\Che-101\menu_example.m
File Edit View Text Debug Breakpoints Web Window Help
z = menu( 'Conversion of Temp', '1. Celsius to Fahrenheit', '2. Fahrenheit to Celsius')
% If Celsius to Fahrenheit is selected a value of 1 is assigned to z.
% If Fahrenheit to Celsius is selected a value of 2 is assigned to z.
```

Option 1

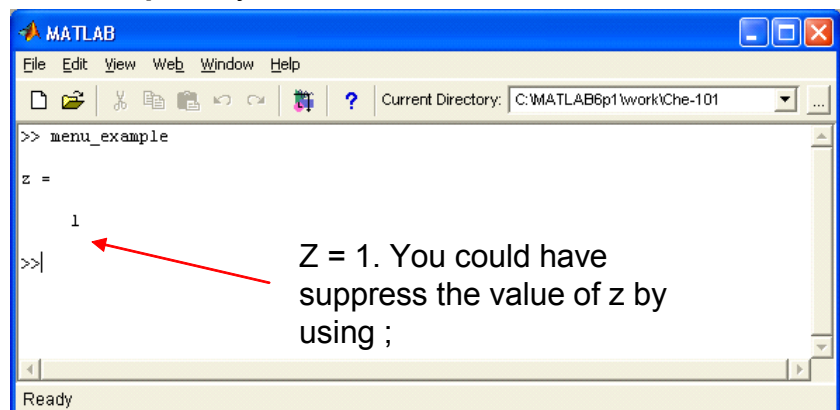
Option 2

Title of your box

Step 2: This is what you will see if you run the program



Step 3: If you select 1. Celsius to Fahrenheit:



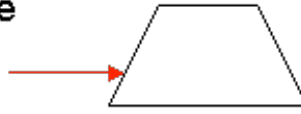
```
MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p1\work\Che-101
>> menu_example
z =
    1
>>
```

Z = 1. You could have suppressed the value of z by using ;

You can use the results from the menu to write a decision loop (if/else syntax).

Display Text or Variable Values on the Screen _ The "disp" Command

This Matlab command is the equivalent to the following unit in a flowchart diagram, see H-2



The "disp" command can be used to display text or the value of a variable without displaying the variable name. Normally the 'fprintf' command is preferred. (see Tutorial V.a).

The syntax for text is: >> disp('text string')

The syntax for variables is: >> disp(variable)

Step 1: Write the code

```

1 %This is an example of the display "disp" command
2 %This command shows what we want on the screen
3 disp ('Hola means hello in Spanish')
  
```

What ever you write in this section will be shown in the screen

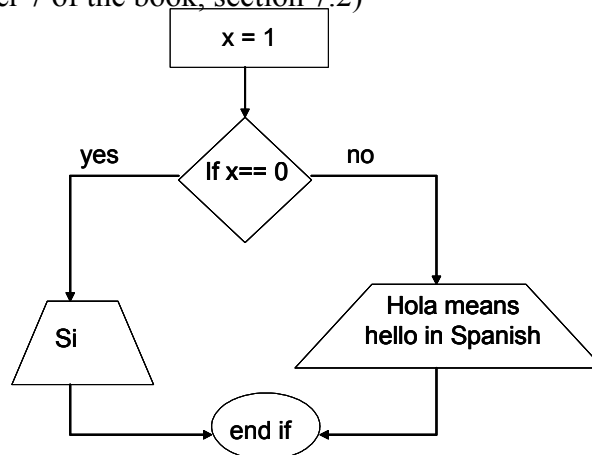
Step 2: Run the program

```

>> exam_disp
Hola means hello in Spanish
>>
  
```

if_end Structures (See Chapter 7 of the book, section 7.2)

This command is equivalent to the following structure in a flowchart diagram (see H-2). The relational operators must be used



This structure causes a sequence of commands to be conditionally evaluated based on a relational test. In the simplest form we decide whether or not to evaluate a list of commands. In more advanced structures we can choose which of many command lists to evaluate.

The simplest if_end structure is:

```
if expression  
    commands  
end
```

The commands are evaluated if all elements of expression are True. See example below:

Step 1: Matlab code

```
1 %This is an example of the use of the if/end structure  
2 x= input ('enter the value for x, use 1 if you want to display message, x = ');  
3 %Use of the if structure  
4 if (x==1)  
5     disp ('Hola means hello in Spanish')  
6 end
```

Step 2: Results from running the program

```
>> exam_disp  
enter the value for x, use 1 if you want to display message, x = 1  
Hola means hello in Spanish  
>> exam_disp  
enter the value for x, use 1 if you want to display message, x = 0  
>>
```

If the user inputs 1 it will print

if expression

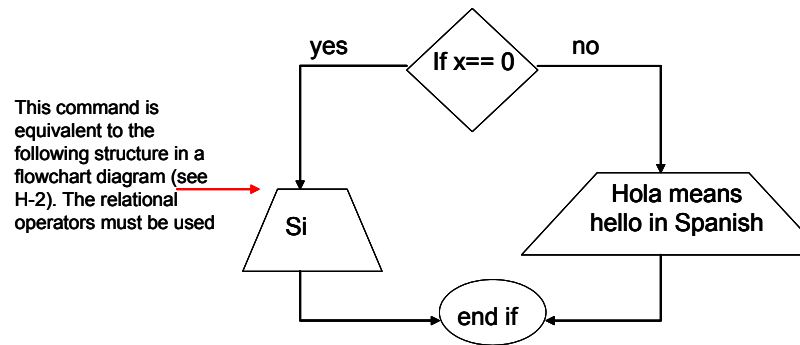
 commands evaluated if expression is True

else

 commands evaluated if expression is False

end

See example below



Step 1: Matlab code

```

1  %This is an example of the use of the if/end structure for two alternatives
2  x= input ('enter the value for x, use 1 if you want to display Hola, x = ');
3  %Use of the if structure with two alternatives
4  if (x==1)
5  disp ('Hola means hello in Spanish')
6  else
7  disp ('Si')
8  end
  
```

Any other value except 1 will display "Si"

Step 2: Running the program

```

>> exam_disp
enter the value for x, use 1 if you want to display Hola, x = 2
Si
>> exam_disp
enter the value for x, use 1 if you want to display Hola, x = 1
Hola means hello in Spanish
>> |
  
```

You can write the same code but using the "menu" command. Try it this way:

```

1  %This is an example of the use of the if/end structure for two alternatives
2  %using the menu command
3  x= menu ('Choices to Print', '1.Hola means hello in Spanish', '2.Si');
4  %Use of the if structure with two alternatives
5  if (x==1)
6  disp ('Hola means hello in Spanish')
7  else
8  disp ('Si')
9  end
  
```

For three or more alternatives:

if expression1

 commands evaluated if expression1 is True

elseif expression2

 commands evaluated if expression2 is True

elseif expression3

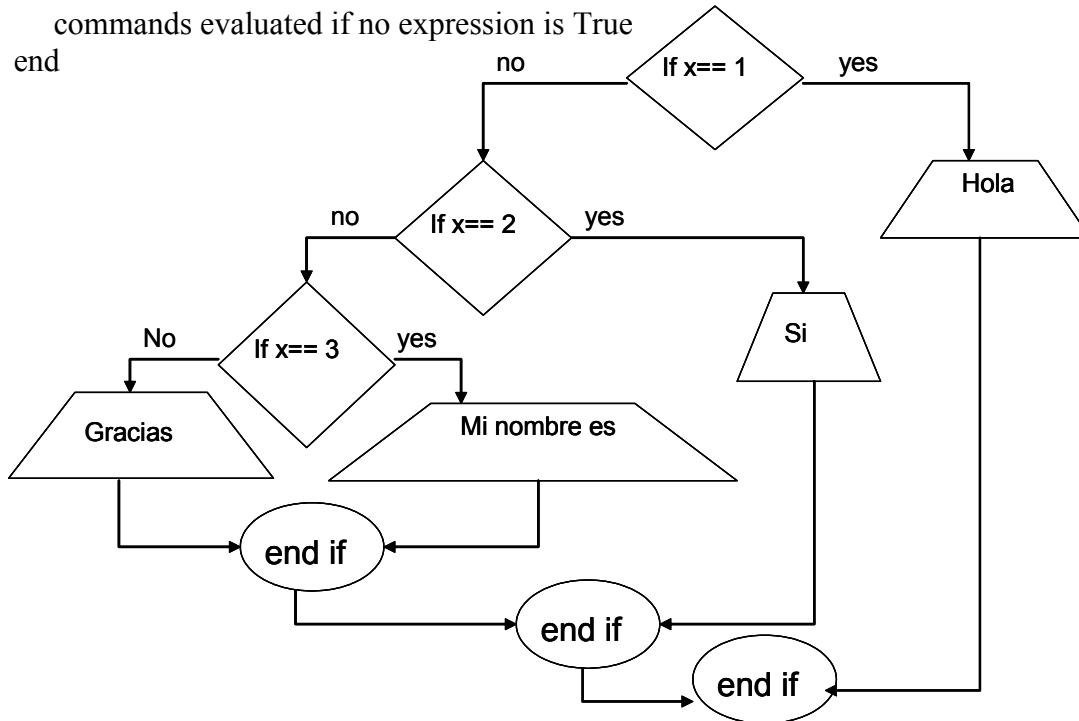
 commands evaluated if expression3 is True

else

 commands evaluated if no expression is True

end

This command is equivalent to the following structure in a flowchart diagram (see H-2). The relational operators must be used



The example below shows how to write a Matlab code for the flowchart shown above:

Step 1: Matlab code

```

C:\MATLAB6p1\work\Che-101\exam_disp.m
File Edit View Text Debug Breakpoints Web Window Help
%This is an example of the use of the if/end structure for several alternatives
%using the menu command
x= menu ('Translate to Spanish the Following Phrases', '1.Hello', '2.Yes', '3.My name is', '4.Thanks');
%Use of the if structure with multiple alternatives
if (x==1)
disp ('Hola')
elseif (x==2)
disp ('Si')
elseif (x==3)
disp ('Mi nombre es')
else
disp ('Gracias')
end

```

Step 2: Running the Code



Step 3: Checking the results

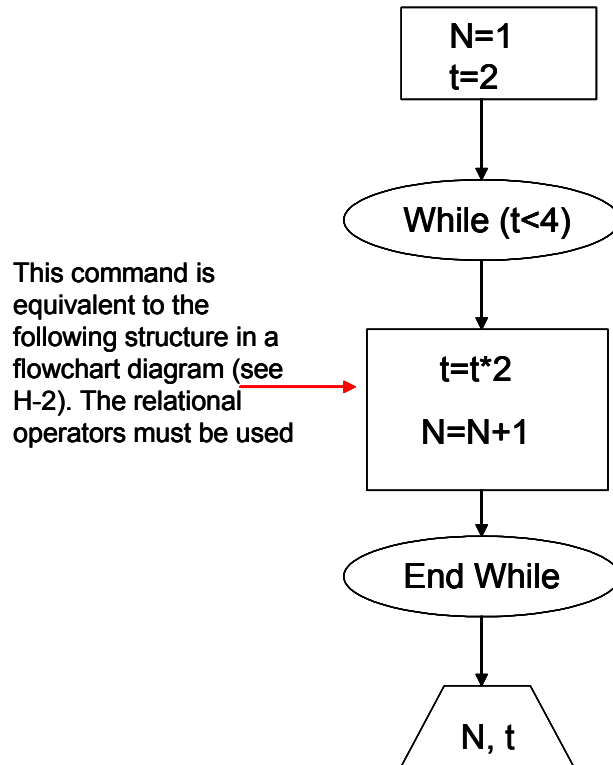
```

MATLAB
File Edit View Web Window Help
Current Directory: C:\MATLAB6p1\work\Che-101
>> exam_disp
Hola
>> exam_disp
Si
>> exam_disp
Mi nombre es
>> exam_disp
Gracias
>>

```

Important: Notice that once you make a choice, your program runs and stops. If you want to make another choice you need to run your program again. If you want to run a certain subroutine in your code (or your whole code) for an indefinite number of times you need to use the “while loop”. This is described in the next section.

while Loops (See Chapter 7 of the book, section 7.4.2)



while loops evaluate a group of commands an indefinite number of times. The general form of the while loop is:

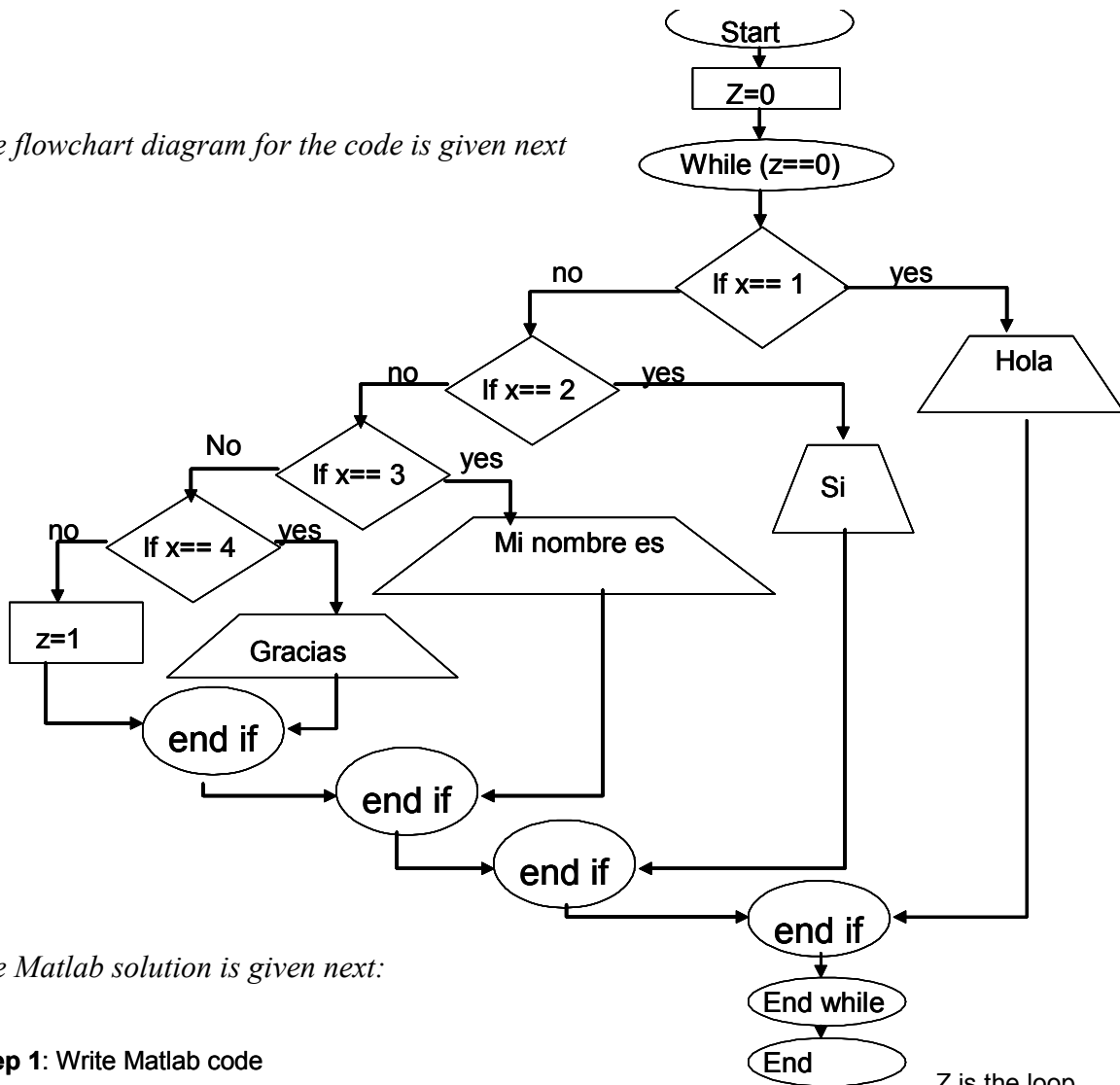
```

while (expression, this is the loop counter/control variable)
  commands
end
  
```

*The commands between the **while** and **end** statements are executed as long as all elements in **expression** are True.*

Example: Modify your translator code (see previous section) to run the program for an undefined number of times. The user must have the choice to exit the program. Use a menu input.

The flowchart diagram for the code is given next



The Matlab solution is given next:

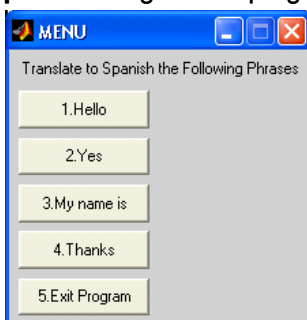
Step 1: Write Matlab code

```

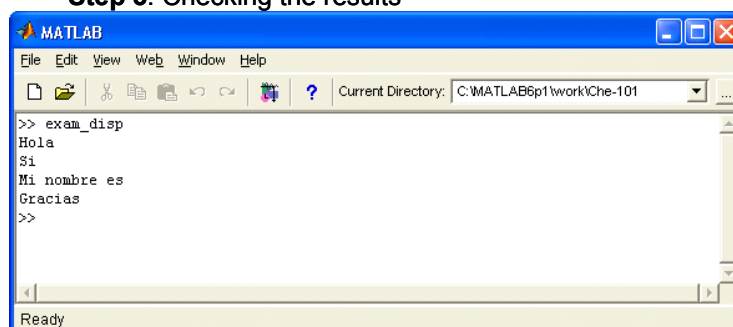
1 %This is an example of the use of while loop with the if/end structure for several alternatives
2 %using the menu command
3 z=0; %The semicolon is used to suppress z on the screen
4 while (z==0) %The program will run until z is different to 0
5 x= menu ('Translate to Spanish the Following Phrases', '1.Hello', '2.Yes', '3.My name is', '4.Thanks', '5.Exit Program');
6 %Use of the if structure with multiple alternatives and the while loop
7 if (x==1)
8 disp ('Hola')
9 elseif (x==2)
10 disp ('Si')
11 elseif (x==3)
12 disp ('Mi nombre es')
13 elseif (x==4)
14 disp ('Gracias')
15 else
16 z=1; %By changing z to 1 the while loop will stop. Remember that z is the loop control variable
17 end %This is the end of the if structure
18 end %This is the end of the while loop
  
```

Z is the loop counter/control variable. The while loop will run until z~0

Step 2: Running Matlab program



Step 3: Checking the results



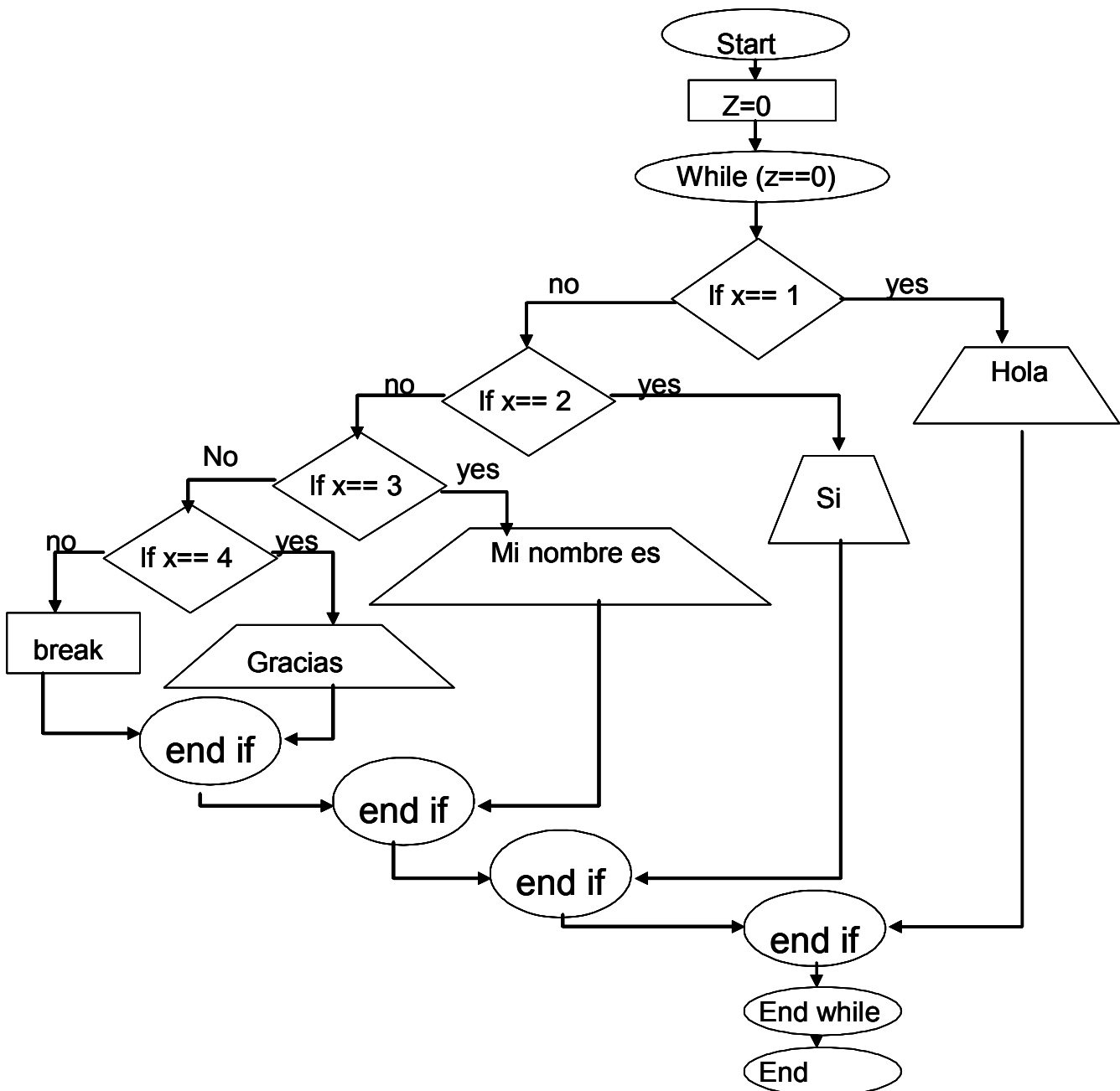
Another option to Break Out of for and while loops

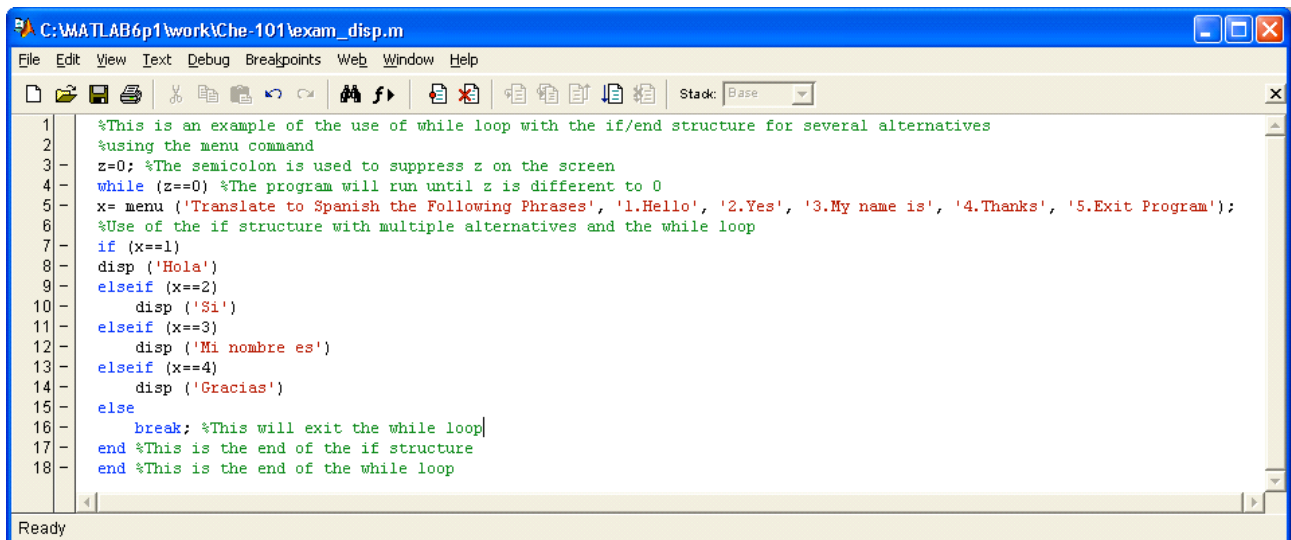
You can jump out of loops by using the **break** command. Inside the loop you would use a section of code looking something like this;

```
break
end
```

If you are not in a loop the **break** command stops execution of the program.

Example, if you modified the example given above using the command break, this will be the flowchart diagram and Matlab code





```
1 %This is an example of the use of while loop with the if/end structure for several alternatives
2 %using the menu command
3
4 z=0; %The semicolon is used to suppress z on the screen
5 while (z==0) %The program will run until z is different to 0
6 x= menu ('Translate to Spanish the Following Phrases', '1.Hello', '2.Yes', '3.My name is', '4.Thanks', '5.Exit Program');
7 %Use of the if structure with multiple alternatives and the while loop
8 if (x==1)
9 disp ('Hola')
10 elseif (x==2)
11     disp ('Si')
12 elseif (x==3)
13     disp ('Mi nombre es')
14 elseif (x==4)
15     disp ('Gracias')
16 else
17     break; %This will exit the while loop
18 end %This is the end of the if structure
19 end %This is the end of the while loop
```

SOLVED PROBLEMS

1. Write a program in Matlab that will allow the user to calculate any of the following information by using ideal gas law: temperature, volume, and pressure. You must use the menu command. The program should run until the user decides, that is, the user must have the choice of exiting the program.

Solution:

1. Follow the “tips for solving problems” discussed in lesson 1
2. Write a flowchart diagram (this was assigned to you in last class, see H-2). In the space given below draw your flowchart diagram

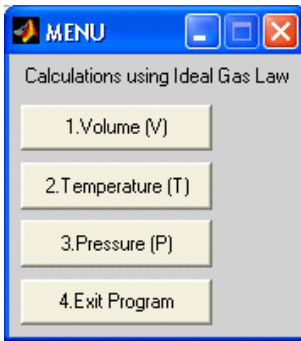
3. Write the code in Matlab. See the solution given below. This is an example of quality program, it includes everything that you must have when you write your codes (notice that the code has comments, displays the purpose of the code and other valuable information, and displays the results with the appropriate units):

```
% This program calculates the temperature, volume, and pressure of a gas using ideal gas law
% Developed by Gerardine Botte
% Created on: 04/06/05
% Last modified on: 04/06/05
% Che-101, Spring 05
% Solution to Solved Problem 1, Tutorial II
% The user will be prompt to input variables depending on his/her choice
%-----
clc %erase the screen
clear %it is convenient to use this to make sure that there are no variables in the memory of matlab

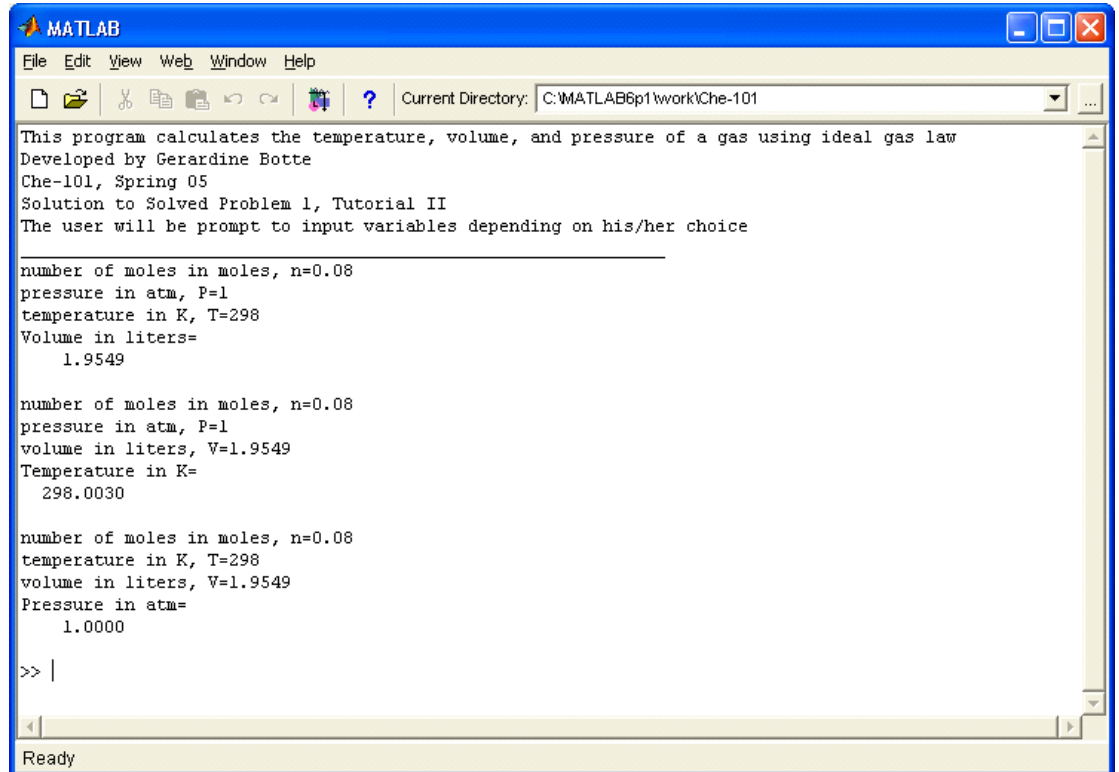
disp('This program calculates the temperature, volume, and pressure of a gas using ideal gas law')
disp('Developed by Gerardine Botte')
disp('Che-101, Spring 05')
disp('Solution to Solved Problem 1, Tutorial II')
disp('The user will be prompt to input variables depending on his/her choice')
disp('_____')

z=0; %This is the loop counter/control variable
R=0.082; %Universal gas constant, 0.082 atm l/(mol K)
while (z==0) %The program will run until z is different to 0
x= menu('Calculations using Ideal Gas Law', '1.Volume (V)', '2.Temperature (T)', '3.Pressure (P)', '4.Exit Program');
%the variable x stores the choice of the user
if (x==1) %This option calculates the volume
    n= input('number of moles in moles, n=');
    P= input('pressure in atm, P=');
    T= input('temperature in K, T=');
    V= n*R*T/P;
    disp('Volume in liters=')
    disp(V)
elseif (x==2) %This option calculates the temperature
    n= input('number of moles in moles, n=');
    P= input('pressure in atm, P=');
    V= input('volume in liters, V=');
    T=P*V/(n*R);
    disp('Temperature in K=')
    disp(T)
elseif (x==3)
    n= input('number of moles in moles, n=');
    T= input('temperature in K, T=');
    V= input('volume in liters, V=');
    P= n*R*T/V;
    disp('Pressure in atm=')
    disp(P)
else
    break; %This will exit the while loop
end %This is the end of the if structure
end %This is the end of the while loop
```

Running the Program



Checking the results



2. The ideal gas law is only valid at low pressures. Write a program in Matlab that will allow the user to calculate the volume of the gas using ideal gas law only if the pressure is lower than 2 atm. You are required to use a while loop to do this (do not use if/else structure). If the pressure is higher than 2 atm the program should print an error message “Pressure higher than 2 atm, ideal gas law is not valid.” The program should run until the user decides, that is, the user must have the choice of exiting the program.

Solution:

1. Follow the “tips for solving problems” discussed in lesson 1
2. Write a flowchart diagram. In the space given below draw your flowchart diagram

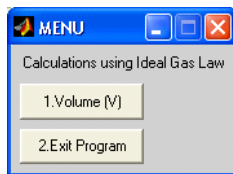
- Write the code in Matlab. See the solution given below. This is an example of quality program, it includes everything that you must have when you write your codes (notice that the code has comments, displays the purpose of the code and other valuable information, and displays the results with the appropriate units):

```

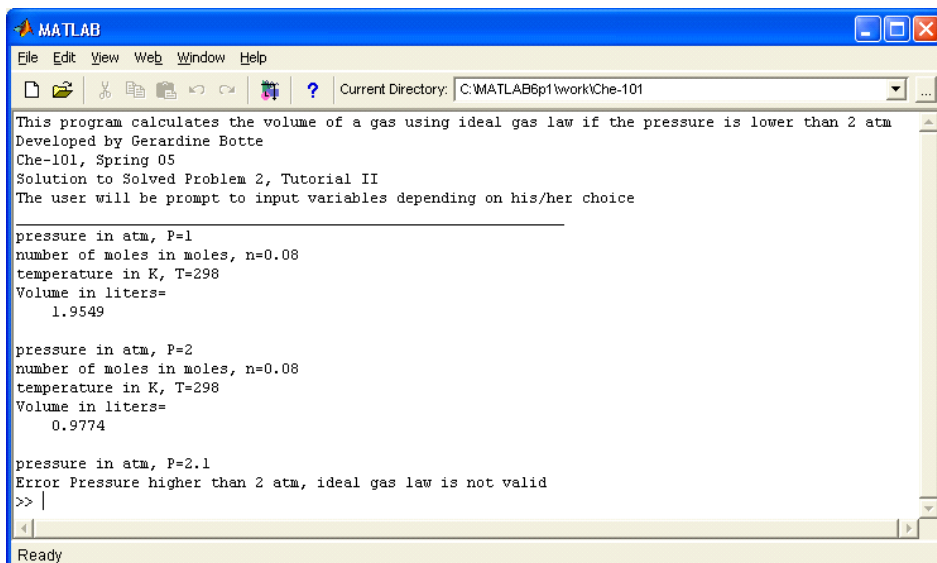
C:\MATLAB6p1\work\Che-101\TII_prob2.m*
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base
1 % This program calculates the volume of a gas using ideal gas law if the pressure is lower than 2 atm
2 % Developed by Gerardine Botte
3 % Created on: 04/06/05
4 % Last modified on: 04/06/05
5 % Che-101, Spring 05
6 % Solution to Solved Problem 2, Tutorial II
7 % The user will be prompt to input variables depending on his/her choice
8 %-----
9 clc %erase the screen
10 clear %it is convenient to use this to make sure that there are no variables in the memory of matlab
11
12 disp ('This program calculates the volume of a gas using ideal gas law if the pressure is lower than 2 atm')
13 disp ('Developed by Gerardine Botte')
14 disp ('Che-101, Spring 05')
15 disp ('Solution to Solved Problem 2, Tutorial II')
16 disp ('The user will be prompt to input variables depending on his/her choice')
17 disp ('-----')
18 z=0; %This is the loop counter/control variable
19 R=0.082; %Universal gas constant, 0.082 atm l/(mol K)
20 while (z==0) %The program will run until z is different to 0
21 x= menu ('Calculations using Ideal Gas Law', '1.Volume (V)', '2.Exit Program');
22 %the variable x stores the choice of the user
23 if (x=1) %This option calculates the volume
24 P= input ('pressure in atm, P=');
25 while (P<=2)
26 n= input ('number of moles in moles, n=');
27 T= input ('temperature in K, T=');
28 V= n*R*T/P;
29 disp ('Volume in liters=')
30 disp (V)
31 break %This will exit the while command since the pressure is not modified inside the loop
32 end
33 while (P>2)
34 disp ('Error Pressure higher than 2 atm, ideal gas law is not valid')
35 break %This will exit the while command since the pressure is not modified inside the loop
36 end
37 else
38 break; %This will exit the while loop
39 end %This is the end of the if structure
40 end %This is the end of the while loop

```

Running the Program



Checking the results



IMPORTANT: The use of a while loop for making decisions as shown above is not the most efficient way to write a code (while loops require more computational time). The author of the program should have used an if/else statement instead.

PROPOSED PROBLEMS

1. Modify solved problem 2 using an if/else statement to make a decision about the value of the pressure. If the pressure is higher than 2 atm the program should print an error message “Pressure higher than 2 atm, ideal gas law is not valid.” The program should run until the user decides, that is, the user must have the choice of exiting the program.

2. The velocity, v , and the distance, d , as a function of time, of a car that accelerates from rest at constant acceleration, a , are given by:

$$v = at \quad d = \frac{1}{2}at^2$$

Write a Matlab program to determine v and d as every second for the first 10 seconds for a car with acceleration of $a = 1.55 \text{ m/s}^2$. The program should be able to print the velocity and distance every 2 seconds. The user must have the choice of exiting the program or recalculating the variables with different accelerations.