

18-Conditional Statements Part 1

text: Chapter 6

ECEGR 101

Engineering Problem Solving with Matlab

Professor Henry Louie



Overview

- Conditional Statements
- If
- If-Else
- If-Elseif



Conditional Statements

- Allow the programmer to make a decision whether to execute a group of commands
- Types of conditional statements:
 - **if-end** structure
 - **if-else-end** structure
 - **if-elseif-else-end** structure
 - **switch** structure



Examples

If I get a raise,
I will buy a new car.

If I get at least a \$100 per week raise,
I will buy a new car;
else,
I will put the money into savings



Examples

If I get at least a \$100 per week raise,
I will buy a new car;
else, if the raise is greater than \$50,
I will buy a new video game;
otherwise,
I will put the money into savings.



The if-end Statement

- Syntax:

```
if logical expression  
    statements  
end
```

- Example:

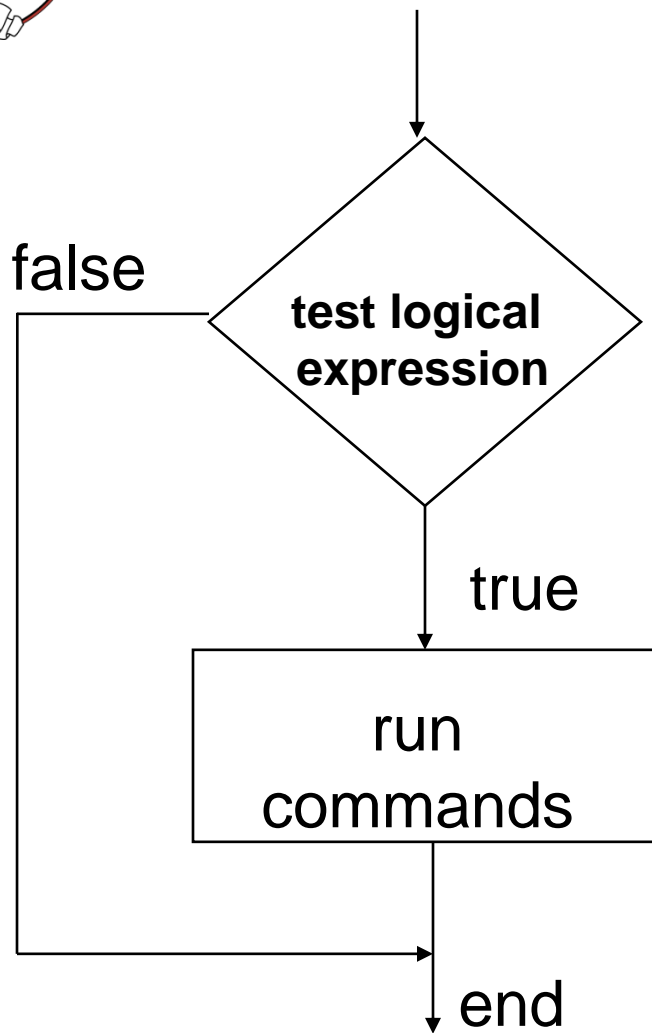
- Suppose that x is scalar and that we want to compute $y = \text{sqrt}(x)$ only if $x \geq 0$.

```
if x >= 0  
    y = sqrt(x);  
end
```

If x is negative, the program takes no action.



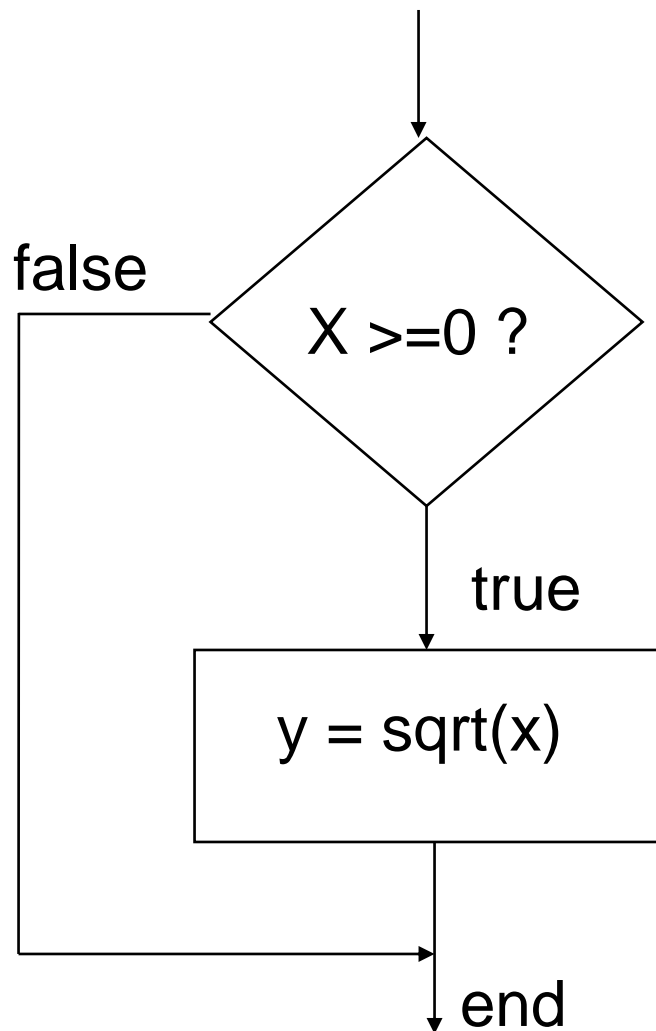
Flowchart for If-End Statement



if logical expression
commands
end



Flowchart for If-End Statement



```
if x >= 0  
    y = sqrt(x);  
end
```




Examples

```
>> x = 9;  
>>  
>> if x >= 0  
y = sqrt(x)  
end  
y =  
    3
```

```
>> clear all  
>> x = -9;  
>>  
>> if x >= 0  
y = sqrt(x)  
end  
>>  
>> y  
??? Undefined function or variable 'y'.
```

```
>> if x >= 0, y = sqrt(x), end  
>>
```



Examples

If x and y have scalar values:

```
z = 0; w = 0;  
if (x >= 0) & (y >= 0)  
    z = sqrt(x) + sqrt(y)  
    w = log(x) - 3*log(y)  
end
```

The values of z and w are computed only if both x and y are nonnegative. Otherwise, z and w retain their values of zero.



Exercise

Write a Matlab script that computes $y = \ln(x)$ if $x > 0$. If $x \leq 0$ then the value of y remains unchanged.



Exercise

Write a Matlab script that computes $y = \ln(x)$ if $x > 0$. If $x \leq 0$ then the value of y remains unchanged.

```
if x>0  
    y = log(x);  
end
```

script saved as
exercise1.m

In the command window:

```
>> format compact
```

```
>> x = 10;
```

```
>> exercise1
```

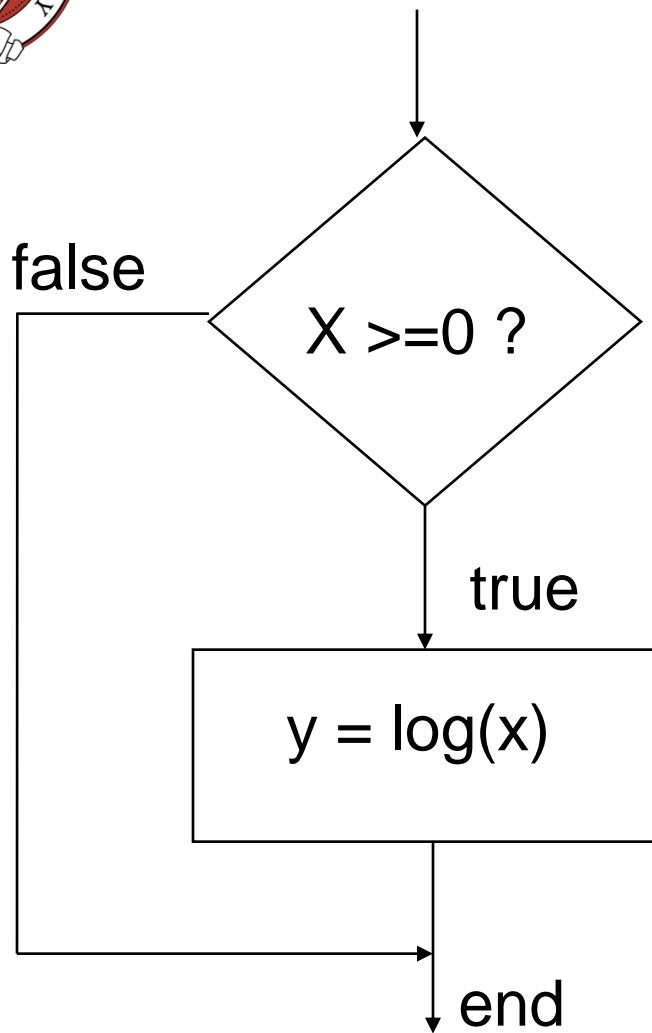
```
>> y
```

```
y =
```

```
2.3026
```



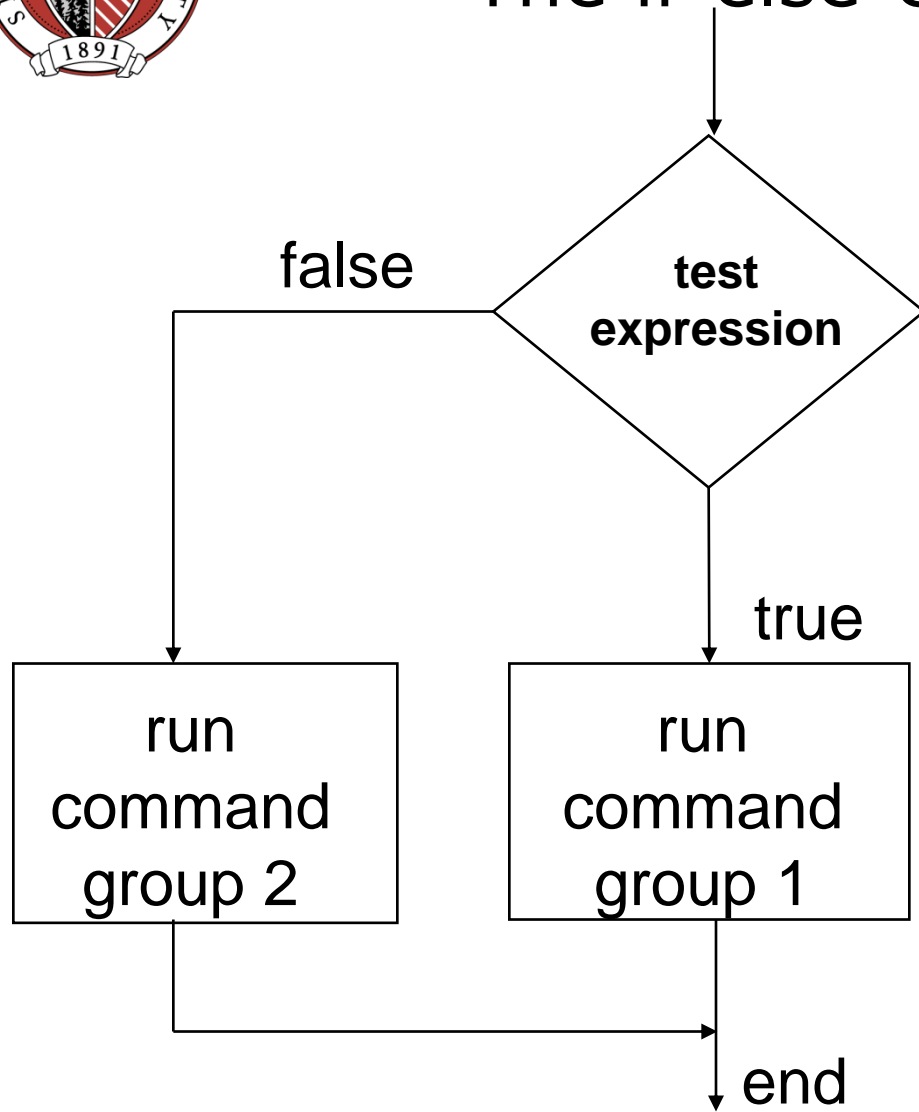
Exercise



```
if x>0
    y = log(x);
end
```



The if-else-end Structure



if logical expression
command group 1
else
command group 2
end

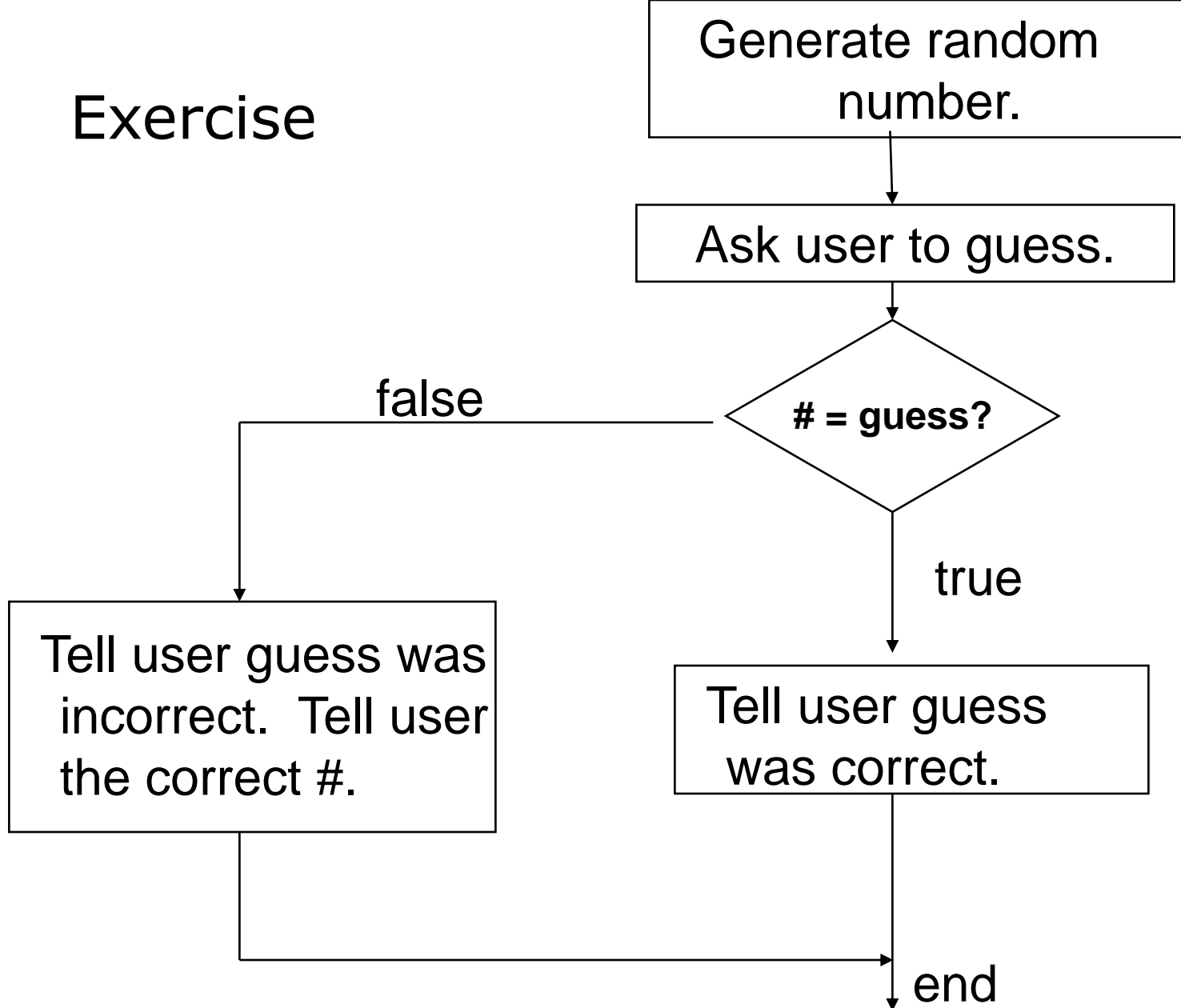


Exercise

Write a program in which the computer generates a **random** integer number between **0** and **10** and then asks the user to guess what the number is. The computer then notifies the user of the result of the guess, and if incorrect, tells the user what the number was.



Exercise





Exercise

```
1 %%Guessing Game
2 - close all; clear all; clc
3 - number=randi(10,1);%%random integer between 1 and 10
4
5 %%User Prompt%
6 - fprintf('I am thinking of a number between 1 and 10.\n')
7 - guess = input('Guess a number: ')
8
9 %%test the number
10
11 - if guess==number
12 -     fprintf('Congratulations-You guessed the correct number. \n')
13 - else
14 -     fprintf('Sorry, your guess was wrong. \n')
15 -     fprintf('The number I was thinking of was %d. \n', number);
16 - end
```



Exercise

```
Command Window
I am thinking of a number between 1 and 10.
Guess a number: 5

guess =

    5

Sorry, your guess was wrong.
The number I was thinking of was 9.
```

```
Command Window
I am thinking of a number between 1 and 10.
Guess a number: 2

guess =

    2

Congratulations-You guessed the correct number.
```

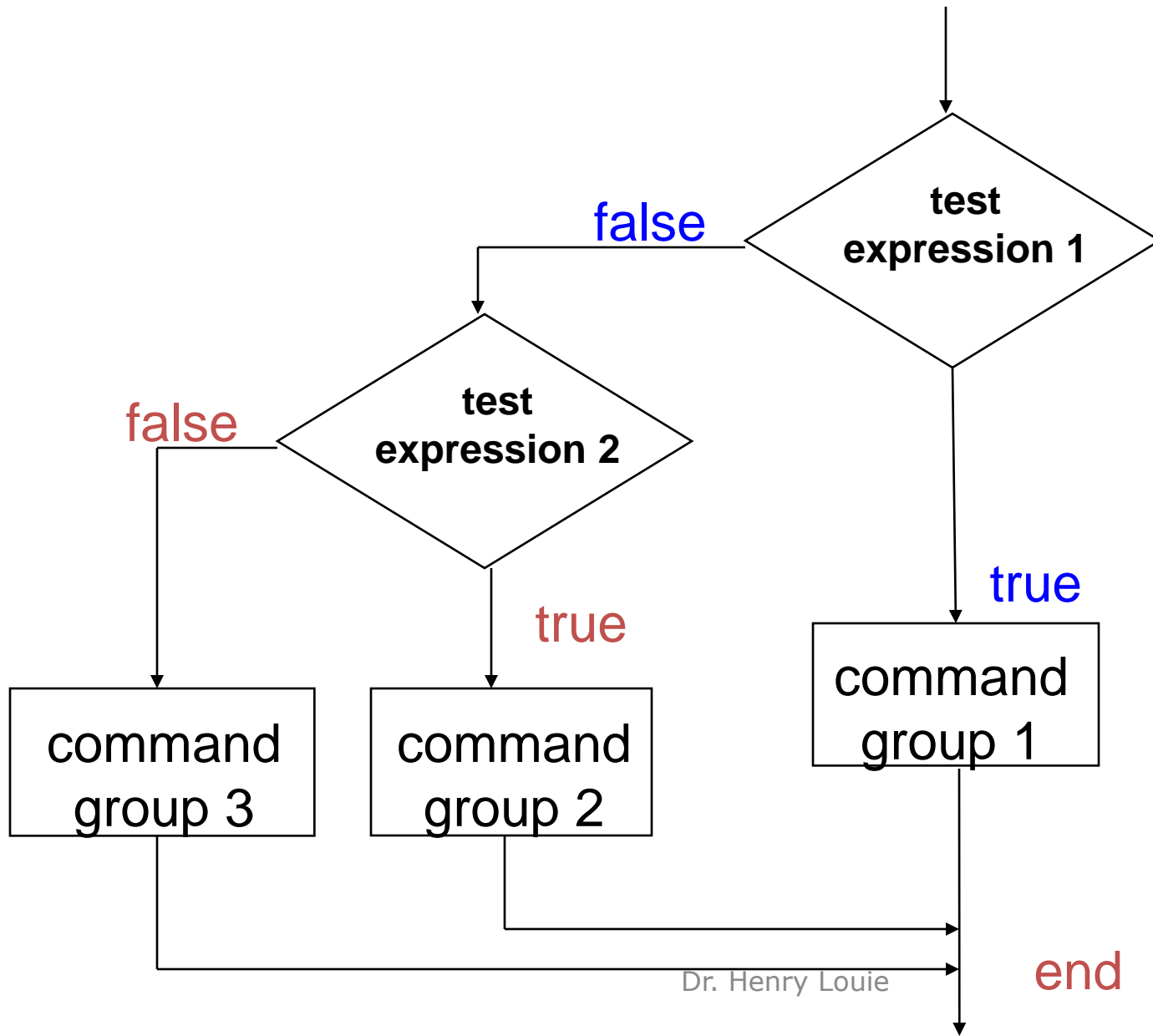


The if-elseif-else-end Statement

Syntax

```
if logical expression 1
    statement group 1
elseif logical expression 2
    statement group 2
else
    statement group 3
end
```

The if-elseif-else-end Statement



```
if logical expression 1  
    command group 1  
elseif logical expression 2  
    command group 2  
else  
    command group 3  
end
```



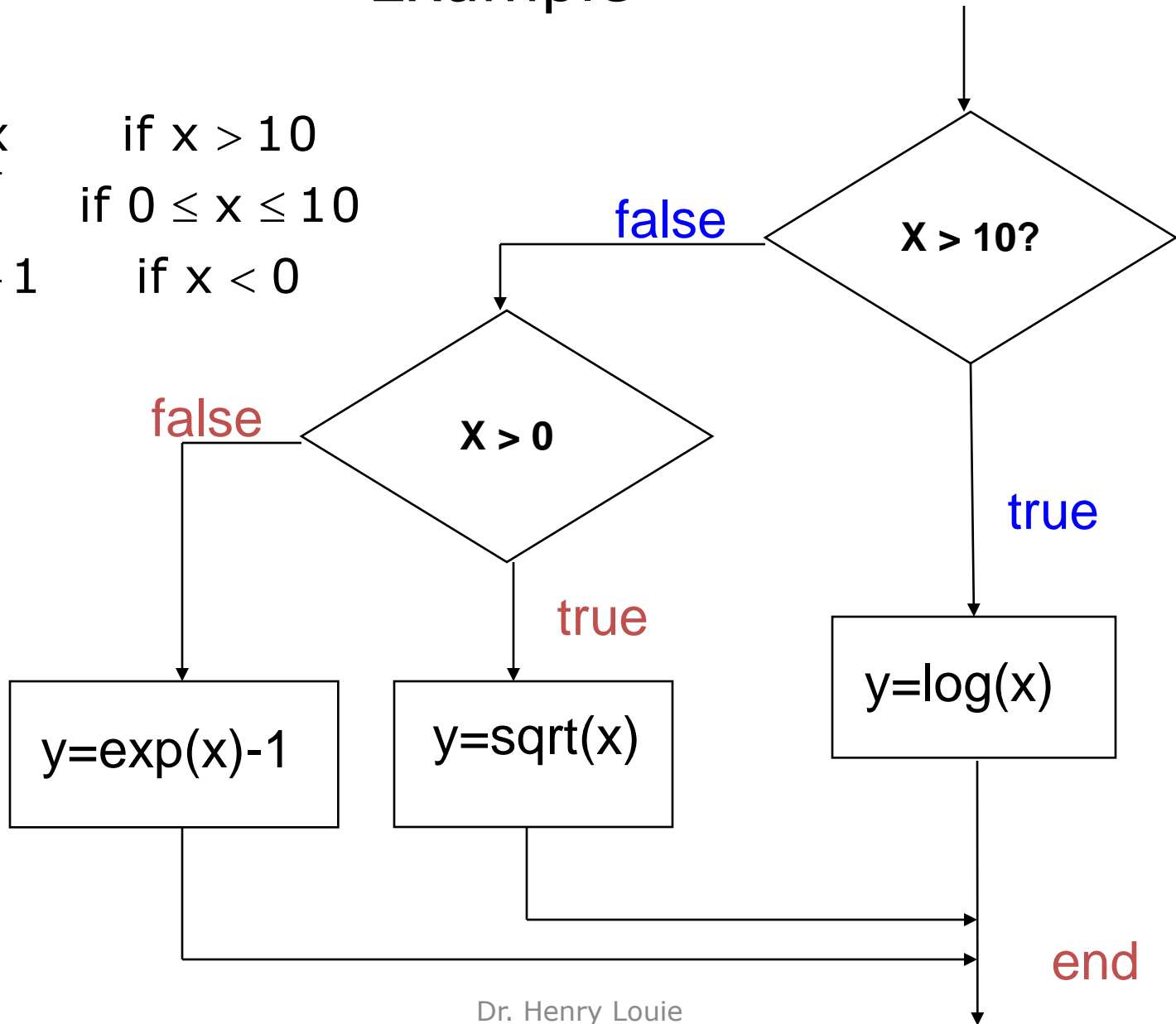
Example

Create a program that asks the user to input a value x , and outputs y based on the following:

$$y = \begin{cases} \ln x & \text{if } x > 10 \\ \sqrt{x} & \text{if } 0 \leq x \leq 10 \\ e^x - 1 & \text{if } x < 0 \end{cases}$$

Example

$$y = \begin{cases} \ln x & \text{if } x > 10 \\ \sqrt{x} & \text{if } 0 \leq x \leq 10 \\ e^x - 1 & \text{if } x < 0 \end{cases}$$





Example

$$y = \begin{cases} \ln x & \text{if } x > 10 \\ \sqrt{x} & \text{if } 0 \leq x \leq 10 \\ e^x - 1 & \text{if } x < 0 \end{cases}$$

```
%%Conditional Statement Example
close all; clear all; clc

%User Prompt%
x = input('Enter X: ')
%test the number
if x>10
    y=log(x)
elseif x >= 0%%we know that x <10, so only need to check if x>=0
    y=sqrt(x)
else%%x is <10, and x<0, so we do not need to check x again
    y=exp(x)-1
end
```

```
Enter X: 14
x =
    14
y =
    2.6391
```

```
Enter X: -4
x =
    -4
y =
   -0.9817
```

```
Enter X: 6
x =
     6
y =
    2.4495
```



Exercise

Consider the following code:

What will be the value of y after this code is executed?

```
x = 82;  
  
if x > 90  
    y = 5;  
elseif x > 80  
    y = 4;  
elseif x > 70  
    y = 3;  
elseif x > 60  
    y = 2;  
end
```




Exercise

Consider the following code:

What will be the value of y after this code is executed?

Answer: $y = 4$.

```
x = 82;  
  
if x > 90  
    y = 5;  
elseif x > 80  
    y = 4;  
elseif x > 70  
    y = 3;  
elseif x > 60  
    y = 2;  
end
```



Exercise

Are the following two programs **equivalent**? That is, do they generate the same result?

```
if x > y
    z = 1;
else
    z = 0;
end
```

```
z = 0;
if x > y
    z = 1;
end
```



Exercise

Are the following two programs equivalent? That is, do they generate the same result?

```
if x > y  
    z = 1;  
else  
    z = 0;  
end
```

```
z = 0;  
if x > y  
    z = 1;  
end
```

Answer: yes.



Example

A, B, and C are logical expressions. Are these program equivalent?

```
if A                                if A
  x = a                              x = a
else                                  elseif B
  if B                                x = b
    x = b                              elseif C
  else                                  x = c
    if C                                else
      x = c                              x = d
    else                                  end
      x = d
    end
  end
end
end
```



Example

A, B, and C are logical expressions. Are these program equivalent?

```
if A                                if A
  x = a                              x = a
else                                  elseif B
  if B                                x = b
    x = b                              elseif C
  else                                  x = c
    if C                                else
      x = c                              x = d
    else                                  end
      x = d
    end
  end
end
end
```

Yes...

```

if x==1
    disp('x is equal to 1')
elseif x==2
    disp('x is equal to 2')
elseif x==3
    disp('x is equal to 3')
else
    disp('x is greater than 3')
end

```

```

if x==1
    disp('x is equal to 1')
else
    if x==2
        disp('x is equal to 2')
    else
        if x==3
            disp('x is equal to 3')
        else
            disp('x is greater than 3')
        end
    end
end
end

```

Are these equivalent for $x \geq 1$?
How about for $x < 1$?

```

if x==1
    disp('x is equal to 1')
else
    if x==2
        disp('x is equal to 2')
    elseif x==3
        disp('x is equal to 3')
    else
        disp('x is greater than 3')
    end
end
end

```

```

if x==1
    disp('x is equal to 1')
elseif x==2
    disp('x is equal to 2')
elseif x==3
    disp('x is equal to 3')
elseif x>=3
    disp('x is greater than 3')
end
end

```



Exercise

Write a program that calculates a student's grade based on three exams and then displays the grade. An exam average:

- greater than or equal to 90 assigns an A
- 89-80 assigns a B
- 79-70 assigns a C
- 69-60 assigns a D
- and less than 60 assigns an F



Exercise

```
%  
% This program requests three exam scores from the user, calculates the  
% average of the three exams, and assigns a grade.  
%  
% Condition          Grade  
%-----  
% Average >= 90      A  
% 80 <= Avg < 90     B  
% 70 <= Avg < 80     C  
% 60 <= Avg < 70     D  
% Average < 60       F  
  
% Ask for the three exam scores.  
exam1 = input('Input Exam Score 1: ');  
exam2 = input('Input Exam Score 2: ');  
exam3 = input('Input Exam Score 3: ');
```



Exercise

```
% Calculate the average exam score
average = mean([exam1 exam2 exam3]);

% Determine the grade
if average >= 90
    grade = 'A';
elseif average >= 80
    grade = 'B';
elseif average >= 70
    grade = 'C';
elseif average >= 60
    grade = 'D';
else
    grade = 'F';
end
```



Exercise

% Print the program result

```
fprintf('Your average exam score is %.2f.\n', average);  
fprintf('This corresponds to a grade of %s.\n\n', grade);
```

Command Window:

```
>> finalGrade  
Input Exam Score 1: 100  
Input Exam Score 2: 90  
Input Exam Score 3: 80  
Your average exam score is 90.00.  
This corresponds to a grade of A.
```

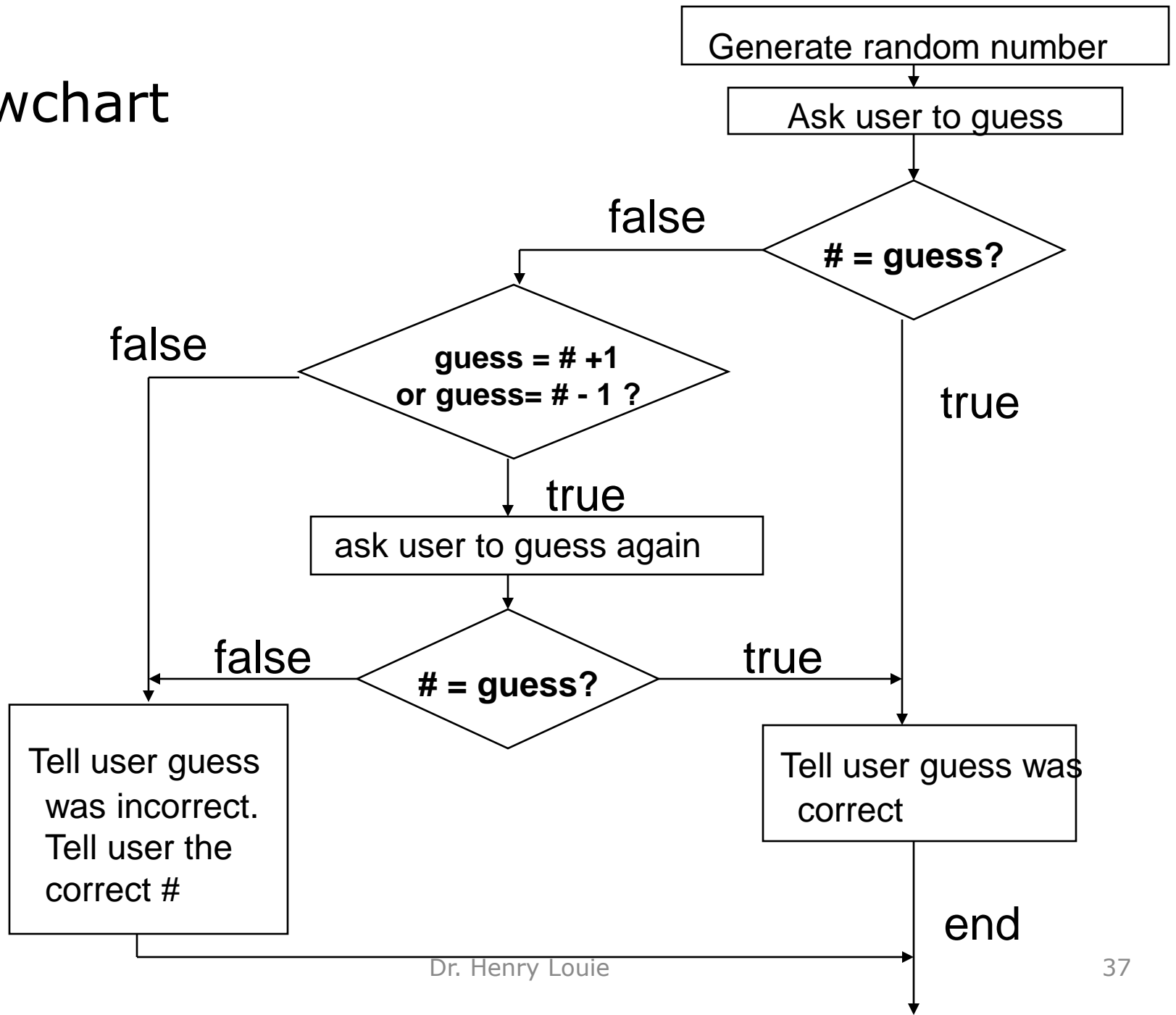
```
>> finalGrade  
Input Exam Score 1: 60  
Input Exam Score 2: 70  
Input Exam Score 3: 95  
Your average exam score is 75.00.  
This corresponds to a grade of C.
```



Exercise

Modify the guessing game program as follows. If the user's guess is **within 1 unit** from the computer's number, reward the user and let him/her **guess again**.

Flowchart





Exercise

```
if (guess = number)
    tell user guess was correct
elseif (guess = number+1)|(guess = number-1)
    tell user the guess was close; ask user to guess again
    if (guess = number)
        tell user guess was correct
    else
        tell user guess was incorrect
        tell user the correct number
    end
else
    tell user guess was incorrect
    tell user the correct number
end
```



Exercise

```
11 % Test the number
12 - if guess == number
13 -     fprintf('Congratulations - You guessed the correct number.\n');
14 - elseif (guess == number+1) | (guess == number-1)
15 -     fprintf('Your guess was wrong but you are really close. Guess again:\n');
16 -     guess = input('Guess a number: ');
17 -     if guess == number
18 -         fprintf('Congratulations - You guessed the correct number.\n');
19 -     else
20 -         fprintf('Your guess was wrong.\n');
21 -         fprintf('The number I was thinking of was %d.\n', number);
22 -     end
23 - else
24 -     fprintf('Your guess was wrong.\n');
25 -     fprintf('The number I was thinking of was %d.\n', number);
26 - end
27
```

```
>> guessing_game2
I am thinking of a number between 1 and 10.
Guess a number: 4
Congratulations - You guessed the correct number.
```

```
>> guessing_game2
I am thinking of a number between 1 and 10.
Guess a number: 3
Your guess was wrong.
The number I was thinking of was 7.
```

```
>> guessing_game2
I am thinking of a number between 1 and 10.
Guess a number: 5
Your guess was wrong but you are really close. Guess again:
Guess a number: 6
Congratulations - You guessed the correct number.
```

```
>> guessing_game2
I am thinking of a number between 1 and 10.
Guess a number: 8
Your guess was wrong but you are really close. Guess again:
Guess a number: 9
Your guess was wrong.
The number I was thinking of was 7.
```




Exercise

Ask the user to input an **integer**. Print "**even**" if the number is even. Print "**odd**" if the number is odd.

Use the `rem` function to test for an even or odd number. The `rem(x,y)` function returns the remainder of dividing `x` by `y`. If `rem(x,2) = 0`, `x` is even.

Use an **if-else** statement in your solution.



Exercise

```
x = input('Enter a number: ');
```

```
if rem(x,2) == 0
```

```
    disp('EVEN');
```

```
else
```

```
    disp('ODD');
```

```
end
```

```
Enter a number: 21
```

```
ODD
```

```
>>
```

```
Enter a number: 44
```

```
EVEN
```

```
>>
```

```
Enter a number: 11.4
```

```
ODD
```

```
>>
```



Exercise

Modify your solution to the previous problem by checking whether the number entered by the user is an **integer**.

One way to do this is to test if the **rounded input is equal to the input**. If it is, then the input is an integer. The **round** function returns the rounded value of the input argument.

If the number is not an integer, print "Not an integer". Use an if-elseif statement in your solution.



Exercise

```
x = input('Enter a number: ');
```

```
if (round(x) ~= x)
```

```
    disp('Not an integer.');
```

```
elseif rem(x,2) == 0
```

```
    disp('EVEN');
```

```
else
```

```
    disp('ODD');
```

```
end
```

```
Enter a number: 34.5
```

```
Not an integer.
```

```
>>
```

```
Enter a number: 3
```

```
ODD
```

```
>>
```

```
Enter a number: 11.467
```

```
Not an integer.
```

```
>>
```