

ENGR 121: Computation Lab I

Handout for Lab 5: Selection Statements

Practice Exercises

1. Write a function called `overtime` that takes `hours` as an input argument. Write an `if` statement within the function that will print “Hey, you get overtime” if the value of `hours` is greater than 40. Test the `if` statement for values of `hours` less than, equal to, and greater than 40.

```
>> overtime(40)
Sorry, you need to work harder to make overtime money.
>> overtime(45)
Hey, you get overtime!
>> overtime(35)
Sorry, you need to work harder to make overtime money.
```

2. Write a function `degorrad` that performs the following tasks:

- Prompt the user for an angle.
- Prompt the user for (r)adians or (d)egrees, with radians to be used as the default.
- If the user enters 'd', the **sind** function will be used to get the sine of the angle in degrees; otherwise the **sin** function will be used. Note that which sine function to use will be based solely on whether the user entered a 'd' or not (that is, 'd' means degrees, so **sind** is used; otherwise, for any other character the default of radians is assumed, so **sin** is used).
- Print the result formatted to two decimal places.
- Return the result back to the calling function or script.

```
>> degorrad;
Please enter the angle: 45
Choose if the angle is in (d)egrees or (r)adians: 'd'
Sin(45) where 45 is in degrees = 0.71.
>> degorrad;
Please enter the angle: 30
Choose if the angle is in (d)egrees or (r)adians: 'r'
Sin(30) where 30 is in radians = -0.99.
>> degorrad;
Please enter the angle: 45
Choose if the angle is in (d)egrees or (r)adians: 'f'
Sin(45) where 45 is in radians = 0.85.
```

3. Write a function called `calcy` that receives a value of x and returns the corresponding value of y . Use the `elseif` clause to choose from among the more than two actions.

$$\begin{aligned} y &= 1 & \text{if } x < -1 \\ y &= x^2 & \text{if } -1 \leq x \leq 2 \\ y &= 4 & \text{if } x > 2 \end{aligned}$$

```
>> x = 1.1;
>> y = calcy(x);
The value of y is: 1.21.
```

4. The following function returns the letter grade corresponding to the quiz grade, which is an integer in the range from 0 to 10.

```
1 function grade = lettergrade(quiz_score)
2
3 % Lets do an bounds check first on the input argument
4 if(quiz_score < 0) || (quiz_score > 10)
5     grade = 'X'; % Letter grade is undefined
6
7 % If we are here in the code, the quiz score is valid
8 elseif quiz_score ≥ 9
9     grade = 'A';
10 elseif quiz_score == 8
11     grade = 'B';
12 elseif quiz_score == 7
13     grade = 'C';
14 elseif quiz_score == 6
15     grade = 'D';
16 else
17     grade = 'F';
18 end % End of the elseif block
19
20 end % End function
```

```
>> grade = lettergrade(20)
grade =
X
>> grade = lettergrade(10)
grade =
A
>> grade = lettergrade(6)
grade =
D
>> grade = lettergrade(5)
grade =
F
```

Rewrite the above function to use the `switch` statement instead of `elseif` to accomplish the same functionality.

5. The Pythagorean theorem states that for a right triangle, the relationship between the length of the hypotenuse c and the lengths of the other sides a and b is given by

$$c^2 = a^2 + b^2$$

Write a script that will prompt the user for the lengths a and c , call a function `findb` to calculate and return the length of b , and print the result. Here is the function:

```
1 function b = findb(a, c)
2 b = sqrt(c^2 - a^2);
3 end
```

Note that any values of a or c that are less than or equal to zero would not make sense; so the script should print an error message if the user enters any invalid value.

6. Write a function `flipvec` that will receive one input argument. If the input argument is a row vector, the function will reverse the order and return a new row vector. If the input argument is a column vector, the function will reverse the order and return a new column vector. If the input argument is a matrix or a scalar, the function will return the input argument unchanged. *Hint:* Use the built-in MATLAB functions, **`fliplr`** and **`flipud`**, in your code to achieve the desired functionality.

```
>> invec = [3 4 7 8 9 2];
>> outvec = flipvec(invec)
The input vector or matrix has 1 rows and 6 columns.
outvec =
     2     9     8     7     4     3
>> invec = [3; 4; 7; 8; 9; 2];
>> outvec = flipvec(invec)
The input vector or matrix has 6 rows and 1 columns.
outvec =
     2
     9
     8
     7
     4
     3
>> invec = [3 4 7; 8 9 2];
>> outvec = flipvec(invec)
The input vector or matrix has 2 rows and 3 columns.
outvec =
     3     4     7
```

```

      8      9      2
>> invec = 3;
>> outvec = flipvec(invec)
The input vector or matrix has 1 rows and 1 columns.
outvec =
      3

```

7. Re-write the following nested if-else statement as a switch statement that accomplishes exactly the same result for all possible values of an integer variable `val`. Assume that “ok”, “xx”, “yy”, “tt”, and “mid” are some user-defined functions. Write the switch statement in the most succinct way.

```

1  %% Using if-else statements
2  val = input('Enter an integer value: ');
3
4  if val > 5
5      if val < 7
6          ok(val)
7      elseif val < 9
8          xx(val)
9      else
10         yy(val)
11     end
12 else
13     if val < 3
14         yy(val)
15     elseif val == 3
16         tt(val)
17     else
18         mid(val)
19     end
20 end

```

8. The Beaufort Wind Scale is used to characterize the strength of winds. The scale uses integer values and goes from a force of 0, which is no wind, up to 12, which is a hurricane. The following script first generates a random force value. Then, it prints a message regarding what type of wind that force represents, using a switch statement. Re-write this switch statement as one nested if-else statement that accomplishes exactly the same thing. You may use else and/or elseif clauses.

```
1 wind_speed = randi([0, 12]);
2 switch wind_speed
3     case 0
4         fprintf('Wind speed = %d. There is no wind. \n', wind_speed);
5     case {1, 2, 3, 4, 5, 6}
6         fprintf('Wind speed = %d. There is a breeze. \n', wind_speed);
7     case {7, 8, 9}
8         fprintf('Wind speed = %d. This is a gale. \n', wind_speed);
9     case {10, 11}
10        fprintf('Wind speed = %d. This is a storm. \n', wind_speed);
11    case 12
12        fprintf('Wind speed = %d. Sharknado! \n', wind_speed);
13 end % End switch-case statement
```