

ENGR 132 Exam Practice

Repetition Structures – While Loops & For Loops

Problems

Practice 1.

For each task described below determine the most appropriate repetitive structure.

- A. You are designing a user-defined function that can calculate the factorial value of a given number (factorial value of a number $N = 1 * 2 * 3 * \dots * N$)
- B. You are determining a solution for your project that requires you to continually ask the user to enter a random number until the user enters a number between 1 and 5.
- C. You are designing a system that will record the data from every sensor in the aircraft.
- D. A manufacturing process requires you to keep the thermostat control within a specified temperature range.
- E. Design a validation process that will iterate through a calculation until a certain level of accuracy is reached.

[Solution](#)

Practice 2.

Consider the following MATLAB code:

```
function prob4(a)
    if a == 151
        for x = 3:-1:1
            z = a*x;
        end
    elseif a == 150
        x = 53;
        y = 98;
        while y < 100
            x = x + 1;
            y = y + 1;
            z = x + 1;
        end
    else
        for y = 2:2:10
            z = z*y;
        end
    end
    fprintf('z = %.0f', z);
```

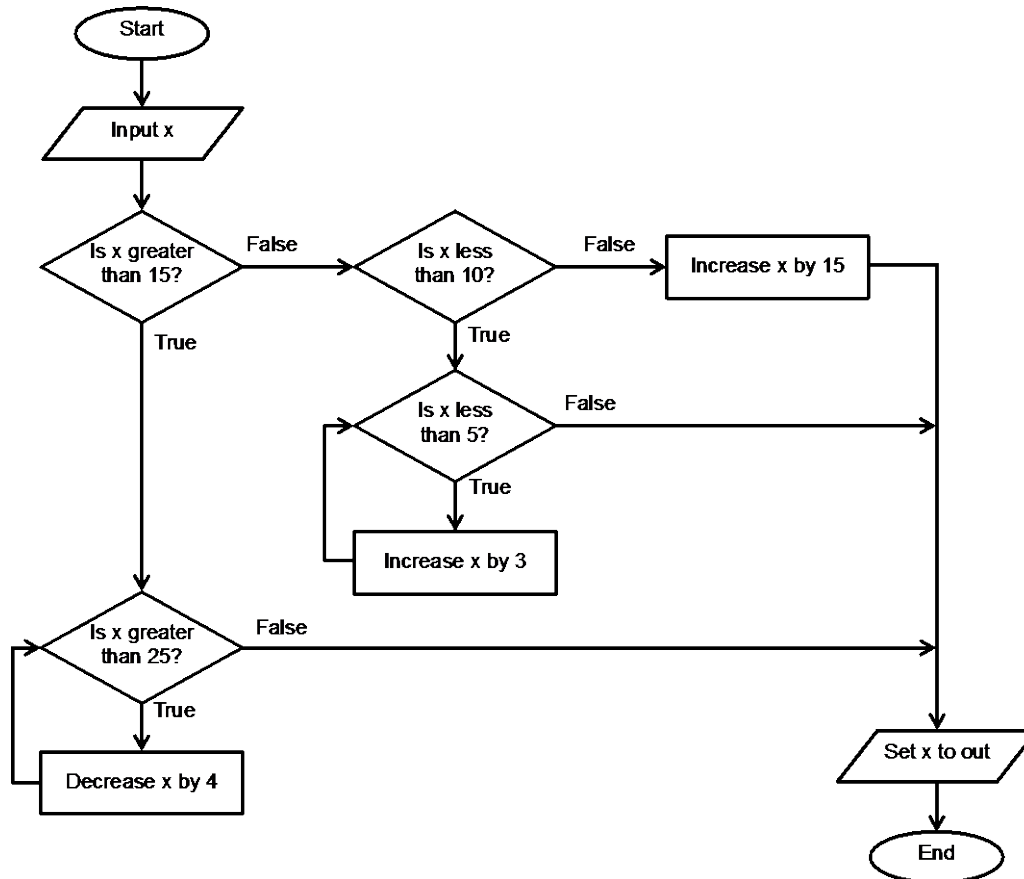
What is displayed in the Command Window after executing each of the following function calls?

- A. >> prob4(151);
- B. >> prob4(150);
- C. >> prob4(152);

[Solution](#)

Practice 3.

Consider the following flowchart:



To help you test your flowchart, your friend Kono gives you two test cases: $x = 12$ and $x = 20$. In addition to these two test cases, come up with two additional test cases (use integers) that test different aspects of the flowchart **that have not been considered by the two test cases already** provided. For each test case, state the purpose of the test case and give the final value of the variable `out`.

	Test Case	Purpose of Test Case	Value of out
1	$x = 12$		out =
2	$x = 20$		out =
3	$x =$		out =
4	$x =$		out =

[Solution](#)

Practice 4.

Replace the **for . . . end** portion of the code (shown in bold) presented below with a single vector operation statement.

```
x = [2 4 6 8 10];  
y = [3 6 9 12 15];  
for k = 1:1:5  
    z(k) = (y(k)*x(k))^2;  
end  
fprintf('%.0f\n', z)
```

[Solution](#)

Practice 5.

A MATLAB program uses the following loop to iteratively perform a simple calculation.

```
count = 0;  
x = 1;  
while x<15  
    count = count + 1;  
    x = x*2;  
end  
fprintf('The value of x is %.0f after %.0f iterations.',x,count)
```

What values of `x` and `count` will be displayed by the `fprintf` statement?

- A. `x` = _____
B. `count` = _____

[Solution](#)

Practice 6.

What output will appear in the MATLAB Command Window when the following code is executed?

```
counter = 36;
while counter >= 28
    counter = counter - 2;
end
fprintf('The value of the counter is %.0f.\n', counter)
```

[Solution](#)

Practice 7.

What output will appear in the MATLAB Command Window when the following code is executed?

```
input_vector = [4 4 9 6];
sum_value = 0;
% fprintf('counter, sum_value\n')
for counter = input_vector
    sum_value = sum_value + counter;
end
fprintf('%.0f, %.0f\n', counter, sum_value)
```

[Solution](#)

Practice 8.

For each type of problem shown below, determine if a **while** loop or a **for** loop is the **most** appropriate repetitive structure. Circle the correct type of loop on the answer sheet.

- A. Finding the square root of a given integer to a specific accuracy level.
- B. Finding the first two hundred numbers of a series.
- C. Recording the data from every fourth sensor along an assembly line.

[Solution](#)

Practice 9.

Consider the MATLAB code below:

```
my_result = 0;
for m = 1:3:10000
    my_result = my_result + m;
end
```

- A. Learn what this code does by tracking the variables through only 4 iterations of the for loop.
- B. Replace the MATLAB code above with one line of code that will produce the same result for my_result.

Answer Sheet section for this problem

A.	Iteration	m	my_result
	0		
	1		
	2		
	3		
	4		
B.			

[Solution](#)

Practice 10.

Consider the MATLAB code below:

What are the values of a, b, c, and d after the following code is executed?

```
a = 2
b = 2
c = 1
while b < a < c
    if b < a
        c = -1
    else
        b = 1
    end
end
d = a/b
```

[Solution](#)

Practice 11.

The `input()` statement in MATLAB is used to request input from users. For example:

```
entered_Value = input('Enter a value such that 1 <= value <= 100')
```

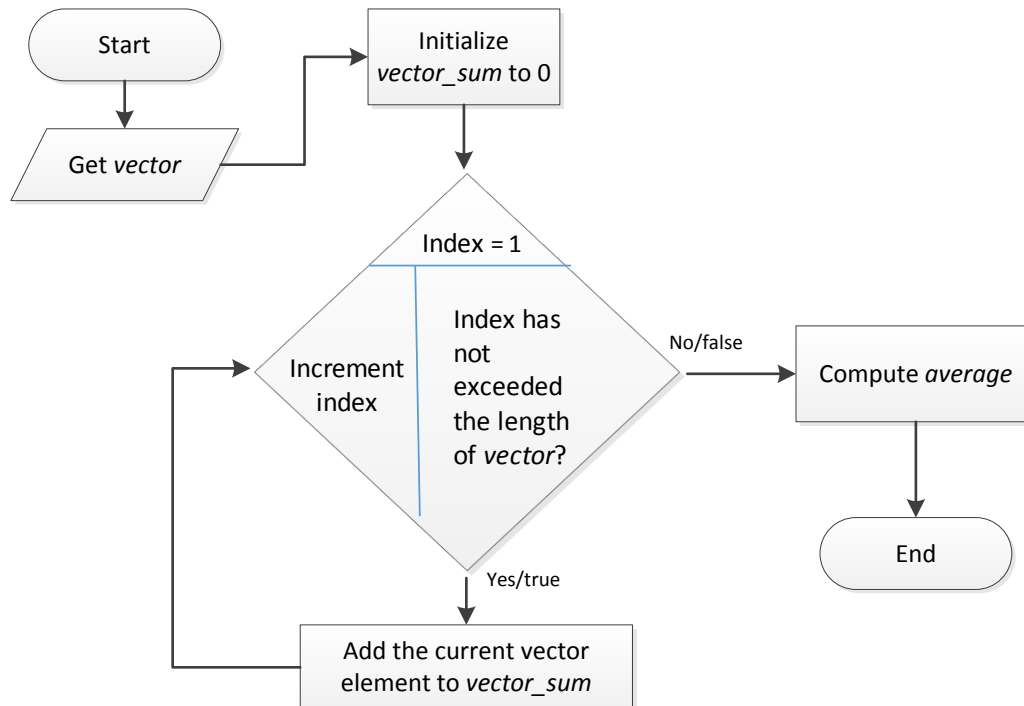
prompts users to enter a value and stores the value entered in the variable `entered_Value`. However, users can enter any value they want and MATLAB has no way of stopping them.

Add code before and/or after the line above to force users to enter values only between 1 and 100. Your code should repeat until the entered value is within the required range, and it should provide a useful warning message each time the input is invalid. Comment your code where clarification is required.

[Solution](#)

Practice 12.

You have been provided with the flowchart for a user-defined function that calculates the average of a vector. The function is named `myAvg`. It accepts the variable `vector` as input and returns the scalar value `average` as output.



Translate this flowchart to MATLAB code.

[Solution](#)

Solutions

Practice Solution 1

- A. `for` loop
- B. `while` loop
- C. `for` loop
- D. `while` loop
- E. `while` loop

Practice Solution 2

- A. `z = 151`
- B. `z = 56`
- C. `Undefined function or variable 'z'.`

Practice Solution 3

	Test Case	Purpose of Test Case	Value of out
1	<code>x = 12</code>	This test case returns false to the first and second decisions of the flowchart, so its value is increased by 12.	<code>out = 27</code>
2	<code>x = 20</code>	This test case returns true to the first decision, but false to the next decision and does not enter the while loop and never changes its value.	<code>out = 20</code>
3/4	<code>x > 25</code>	Test cases of these values will return true to the first decision and the next decision will enter the while loop.	<code>out = 25, 24, 23, or 22</code> (Based on initial x)
3/4	<code>5 <= x < 10</code>	Test cases of these values will return a false to the first decision, a true to the second decision, but false to the third decision and will not enter the while loop. They will not change their value.	<code>out = 5, 6, 7, 8, or 9</code> (Same as initial x)
3/4	<code>x < 5</code>	Test cases of these values will return a false to the first decision, a true to the second and third decisions and enter the while loop.	<code>out = 5, 6, or 7</code> (Based on initial x)

Practice Solution 4

```
x = [2 4 6 8 10];
y = [3 6 9 12 15];
z = (y.*x).^2
fprintf('%.0f\n', z)
```


Practice Solution 5

The value of x is 16 after 4 iterations.

or

x = 16

Count = 4

Practice Solution 6

Answer: The value of the counter is 26.

Practice Solution 7

counter	sum_value
6	23

Practice Solution 8

- A. while
- B. for
- C. for

Practice Solution 9

A.	Iteration	m	my_result
	0	--	0
	1	1	1
	2	4	5
	3	7	15
	4	10	22
B.	my_result=sum([1:3:10000])		

Practice Solution 10

- a. 2
- b. 1
- c. 1
- d. 2

Practice Solution 11

```
entered_Value = input('Enter a value such that 1<= value <= 100: ');
while entered_Value > 100 | entered_Value < 1 OR
    ~(entered_Value <= 100 & entered_Value >= 1) OR other equivalent logic
    fprintf('Invalid input.\n')
```

```
    entered_Value = input('Enter a value such that 1<= value <= 100: ');  
end
```

Practice Solution 12

```
function average = myAvg(vector)  
  
% Initialize vector_sum to 0  
vector_sum = 0;  
  
% Iterate through vector  
for count = 1:1:length(vector)  
    vector_sum = vector_sum + vector(count);  
end  
  
% Calculate the average  
average = vector_sum/count;
```