

Engineering Analysis I, Fall 2023

Midterm 1

SOLUTIONS

Section number _____

Section number	Discussion time	Instructor
22	8:00 a.m.	Prem Kumar
23	9:00 a.m.	Ilya Mikhelson
24	10:00 a.m.	Prem Kumar
25	11:00 a.m.	Michael Honig
27	12:00 p.m.	Michael Honig

This exam is closed-book and closed-notes. Calculators, computers, phones, or other computing/communication devices are not allowed.

Students should skip this page—it is only for graders.

Question	Points	Score
1	30	
2	20	
3	25	
4	25	
Total:	100	

Answer each question in the space provided. There are 4 questions for a total of 100 points.

1. Put a check mark ✓ in the box next to **EACH** correct answer. Note that there may be more than one correct answer for each question!

- (a) [6 points] Which of the following six MATLAB statements will create (or overwrite) a variable **A** and assign it the matrix $\begin{bmatrix} 1 & 4 & 7 \\ 5 & 3 & 1 \end{bmatrix}$?

✓ <code>A = [[1,4],7; 5,[3,1]]</code>	<input type="checkbox"/> <code>A = [1 4 7]; [5 3 1]</code>
✓ <code>A = [1 4 7; 5 3 1]</code>	✓ <code>A = [1:3:7; 5:-2:1]</code>
✓ <code>A = [1 5; 4 3; 7 1]'</code>	<input type="checkbox"/> <code>[1 4 7; 5 3 1] = A</code>

- (b) [4 points] Which of the following four MATLAB statements will replace the third row of an existing matrix variable **A** with the sum of its first and second rows? Assume that **A** is a 4 by 4 square matrix.

<input type="checkbox"/> <code>A(:,3) = A(:,1) + A(:,2)</code>	✓ <code>A(3,:) = A(2,:) + A(1,:)</code>
<input type="checkbox"/> <code>A(3,end) = A(1,end) + A(2,end)</code>	✓ <code>A(3,1:size(A,2)) = ... A(1,1:end) + A(2,1:end)</code>

- (c) [4 points] Which of the following four MATLAB statements *will* generate an error message?

✓ <code>x = [1 3]^[2 1]</code>	✓ <code>x = [7 9] . = 8</code>
<input type="checkbox"/> <code>x = 1==2</code>	<input type="checkbox"/> <code>x = cos(1:0.1:10)</code>

- (d) [4 points] Which of the following four MATLAB logical expressions will return a value of logical 0 (meaning “false”)?

☐ `(~(1>2)) | (3>2)`

✓ `(1<3) && (4>=9)`

☐ `(1>2) | (3>2)`

✓ `(2<1) || (3~=3) || (~ (4>=1)) || ...
('a' == 'b')`

- (e) [6 points] Which of the following six MATLAB statements is a *not* a valid first line of a loop? Assume that the variable `k` already exists in the workspace and contains the row vector `[1 1 1]`.

☐ `for k = 2:6`

✓ `while k = 1:3`

☐ `for ii = k`

✓ `for 0 < k < 4`

✓ `while k = [1 1 1]`

✓ `while k = k + 2`

- (f) [6 points] Which of the following six blocks of code will double each element of an existing (and possibly non-square) matrix `A`? Note that the doubled values should be accessible after the calculations are performed.

☐ `A = (zeros(size(A))+2)*A`

☐

```
for ii = 1:size(A)
    A(ii,ii) = A(ii,ii)*2;
end
```

✓ `A = A.*2;`

✓

```
for ii = 1:size(A,2)
    jj = 1;
    while jj <= size(A,1)
        A(jj,ii) = A(jj,ii)*2;
        jj = jj + 1;
    end
end
```

✓ `A = A*2;`

✓ `A = A.*ones(size(A))*2;`

2. Suppose each section of code below is run in Matlab. If Matlab generates an error message for the given code section, write “error” on the associated line. Otherwise, write the value that the variable `x` will have after the code section is run.

(a) [4 points]

```
clear; x = 1; y = 3;
if x == 1 && y == 6
    x = x - 1;
elseif x > 1 || y <= 3
    x = x + 2*y;
elseif x >= 1
    x = 22;
else
    x = 44;
end
```

(a) 7

(b) [4 points]

```
clear;
for y = 1:5
    x = y + y^2;
end
```

(b) 30

(c) [4 points]

```
clear;
for y = 1:10
    x = x + y^2;
end
```

(c) error

(d) [4 points]

```
clear;
for y = 0:2:10
    if y == 0
        x = 3;
    else
        x = x + 1;
    end
end
```

(d) 8

(e) [4 points]

```
clear; x = 0;
while x <= 16
    x = x + 4;
    y = x - 4;
end
```

(e) 20

3. [25 points] Complete the MATLAB script below, which simulates rolling 10 six-sided dice and estimates the probability of getting *exactly* 3 repetitions of the first dice value. That is, compute the fraction of trials (groups of 10 rolls) in which the value of the first die appears a total of 3 times across the 10 dice. **Complete each statement with a blank line.**

```
1 % Prompt user to input the number of trials.
2 n = input('Enter number of trials: '); (2 points)
3
4 % Initialize the counter that counts the number of times
5 % 3 repetitions occur.
6 counter = 0; (2 points)
7
8 % Generate n rolls, count the number of times
9 % the first value appears 3 times
10
11 for ii = 1:n
12     % Generate a 1 by 10 array with random integers between 1 and 6.
13     roll = randi(6,1,10); (3 points)
14
15     % First roll value.
16     first_roll = roll(1);
17
18     % Count the number of repetitions of the first value.
19     reps = 0;
20
21     for jj = 2:10 (3 points)
22         if roll(jj) == first_roll (3 points)
23             reps = reps + 1; (2 points)
24         end
25     end
26
27     if reps == 2; (3 points)
28         counter = counter + 1; (2 points)
29     end
30 end
31
32 probability = counter/n; (3 points)
33 disp(probability) (2 points)
```

4. [25 points] For this question, we will find the probability of two people sharing the same birthday in a room of 50 people. To find this, we will perform an experiment. We will approach random people on the street and ask their birthday, and keep a list of all of the birthdays we have encountered. If a new person's birthday is already on our list, we end the experiment and note how many people we had to ask in order to encounter a repetition. We will repeat this experiment N times. At the end, we will see how many experiments required asking 50 people or fewer, and how many required asking more. Using this information, we can then estimate the desired probability as:

$$\text{Probability} = \frac{\text{Number of trials where we had to ask 50 people or fewer before a repetition}}{\text{Total number of trials}}$$

Fill in the blanks below to complete this task.

```
% Ask the user for the number of trials.
num_trials = input('Enter the number of trials: ');           (2 points)

% Allocate a vector to store the number of people for each trial.
results = zeros(num_trials,1);

% Iterate through all of the trials.
for ii = 1:num_trials
    % Initialize a counter to count number of people until a match.
    counter = 0;                                           (2 points)
    % Initialize a variable to keep track of when a match is found.
    match_found = false;                                   (2 points)
    % Initialize a list of birthdays you have come across.
    birthdays = [];                                       (2 points)
    % Keep asking people until you find a match.
    while ~match_found
        % Get a new birthday. (assume only 365 days in the year)
        new_birthday = randi(365);                         (2 points)
        % Check if new birthday is already on the list.
        if sum(birthdays == new_birthday) > 0             (5 points)
            % Stop the trial.
            match_found = true;
        else
            % Add the birthday to the list.
            birthdays = [birthdays; new_birthday];
        end
        % Increment the counter to keep track of number of people.
        counter = counter + 1;
    end
    % Fill in the results vector with the number of people you asked.
    results(ii) = counter;
```

```
end

% Find the number of trials where we had to ask 50 or fewer people.
num_50_or_under = sum(results <= 50); (5 points)

% Calculate the probability as stated in the problem description.
prob = num_50_or_under / num_trials; (3 points)

% Print 'The probability is X', where X is replaced with the
% probability you found to five decimal places.
fprintf ('The probability is %.5f', prob) (2 points)
```