
(Print last name, first name, middle initial/name)

(Student ID)

Statement of integrity: I did not, and will not, break the rules of academic integrity on this exam:

(Signature)

Instructions:

- Read each problem *completely* before starting it!
- Do not use calculators, reference sheets, or any other material. This test is closed book.
- Solve each problem using MATLAB, except where indicated.
- Use only specified code in each problem.
- Write your solutions directly on the test using blue/black pen or pencil. Clearly indicate which problem that you are solving. You may write on the back of each sheet. If you need scrap paper, ask a proctor.
- Provide only *one* statement, expression, value, or comment per blank!
- Do *not* alter, add, or remove any code that surrounds the blanks and boxes.
- Do *not* supply multiple answers. If you do so, we will grade only one that we will choose.
- Show all work, especially algorithms. Better that you explain how you would solve a problem than to leave it blank.
- Follow good style! When possible, keep solutions general, avoid redundant code, use descriptive variable names, use named constants, indent substructures, avoid breaking out of loops, and maintain other tenets of programming philosophy.
- Comment each control structure and major variable, *briefly*.
- Do not dwell on a problem if you get stuck. Do the other problems first!
- Raise your hand if you have any questions.

Points:

1. _____ (20 points)

2. _____ (20 points)

3. _____ (40 points)

Subtotal: _____/(100 points)

Bonus _____ (___ points)

Total: _____/(100 points)

Problem 1 [20 points] *Nested loops, Character graphics*

Task: Write a program that prints a *trapezoid* of stars ('*') for a depth supplied by the user. This trapezoid will always have three stars on top. For instance, the following trapezoid has a depth of five:

```

  ***
 *****
 *******
 *****
  ***

```

Approaches: There many ways to do this problem. You might wish to follow this algorithm:

- Start with a row.
- For the current row, print the columns.
 - Print a space until you reach “the end,” which varies as the difference **depth-row**.
 - Print stars until you reach spaces again.
- Go to the next row until you exceed the depth.

Specifications: Do not use arrays, use nested control structures, and try to be general when possible. You may create and use any variables that you need.

```

% Problem1
% Get user input:
depth = str2double(input('Enter the depth: ','s'));
while ~isreal(depth) | isinf(depth) | isnan(depth) | ~isnumeric(depth) | ...
    floor(depth)~=ceil(depth)|depth < 1
    disp(['Please enter a legal integer!']);
depth = str2double(input('Enter the depth: ','s'));
end

% Print trapezoid of *s, assuming three stars on top:

```

```

end

```

Problem 2 [20 points] *Arrays*

Complete the following program that sums the columns of a given array called **values** and stores the results in a 1D array called **sumCols**. You must use nested loops. For bonus points, below the problem show how you can solve the problem without using loops.

```
% Problem2
```

```
% Example array to use:
```

```
values = [1 2 3; 4 5 6; 7 8 9];
```

```
% Get rows and cols of the array:
```

```
_____ = size(values);
```

```
% Create a 1D array of zeros to store the results):
```

```
sumCols = zeros(1,rows);
```

```
% Sum the columns of the array and store the results in sumCols:
```

```
% Output the values in sumCols:
```

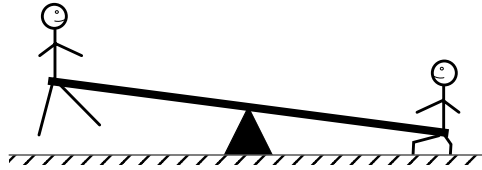
```
_____
```

```
% Bonus solution:
```

```
_____
```

Problem 3 [40 points] *Simulation, Loop design, algorithms*

Background: Suppose that you want to simulate the operation of a *see-saw*, as shown below. Two people push each other up and down, which causes them to burn calories. Assume that the board starts in the bottom position on the right. One cycle of operation consists of the board going up and down again, which consumes ten calories per person.



The people will play for twenty-five cycles, unless one of the people gets bored, which happens *about* 10% of the time. Check for boredom at the beginning of each cycle, including the beginning of the session.

Goal: Determine and report how many calories the people burn in one session.

Tasks: You will write an algorithm for the simulation. After writing the algorithm, you need to write the program based on the algorithm.

Requirements: You will need to use the function `rand`, which returns a random value between 0 and 1. Do not use arrays.

Algorithm [15 points]:

Solution [25 points]:

Checklist: Congratulations! You reached the last page of Prelim 2. Make sure that you clearly indicate your name, ID, and section. Also, re-read all of the problem descriptions/code comments/instructions. If you reached this part before exhausting the allotted time, check your test! Maybe you made a simple mistake? You should check the following:

- ___ maintained all assumptions
 - ___ remembered punctuation, such as semicolons and braces
 - ___ didn't confuse *equals* with *assign* operators
 - ___ completed all tasks
 - ___ filled in ALL required blanks
 - ___ given comments when necessary
 - ___ declared all variables
 - ___ maintained case-sensitivity
 - ___ handled "special cases" correctly
 - ___ indicated which solution to grade if you wrote multiple attempts
-

Bonus problems: Do these only if you have completely finished the other problems:

1) What is the secret number for this bonus question that was given in lecture to reward those people who came?

2) What is the syntax of a function?

3) What is the output from the following statements?

```
>> a=[1:3; 4:6];  
>> a([3:-1:2],[2 1 3]) = a;
```