

EXAM 2 – Computer PORTION

For this exam, you will create FOUR m-files. A main program called `lastname_firstname.m` (all lower case) and 3 functions with names specified below. The header section of your code should include your name, section number, and CM number; you do not need to do any other commenting. There should be no output other than what is requested.

When you are done, email all four of your m-files to your instructor.

NOTE: All programming must stop at 8:30 pm. You will have a few minutes after that to email your files if you need that time.

Problem (60 pts)

a) (20 points) For this part, you must write a function that will plot a circle using symbols.

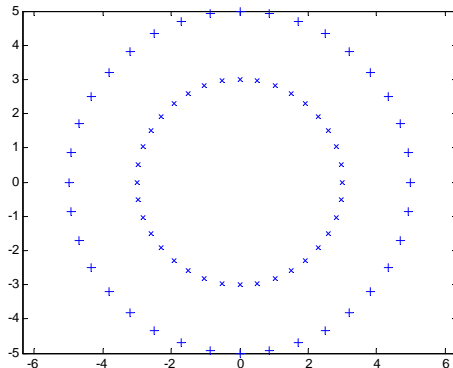
Your function should have the following form:

```
lastname_circle(R, symbolswitch)
```

- R is the radius of the circle.
- If `symbolswitch` is a 1, then your function should plot a circle using “x” symbols. If `symbolswitch` is a 2, then your function should plot a circle using “+” symbols.
- Your circle should include 36 symbols and be centered at $(x,y) = (0,0)$.
- (Hint: Recall that the equation for a circle is $x = R \cos \theta$ and $y = R \sin \theta$)

Your main program should call your function twice:

- Call the function to make a circle of radius 3 using “x” symbols
- Call the function to make a circle of radius 5 using “+” symbols
- Both circles should appear on the same figure.
- (Hint: Use `axis equal` to make the circles appear properly round.)

**TURN THE PAGE OVER**

- b) (20 points) The Mars Curiosity Rover can only generate 2.5 kW-hrs of energy per day, so you must determine how many tasks the rover can do in one day. Once again, you will create a function and call it from your main program.

Your main program should:

- Load the data in the Excel file called `rover_tasks.xlsx`
- Call your function with the first column of `rover_tasks`, then the second column of `rover_tasks`, then the third column of `rover_tasks`
- Print the results to the command window:

```
From column 1, the first 37 tasks can be done.  
From column 2, the first 66 tasks can be done.  
From column 3, the first 52 tasks can be done.
```

Your function call should have the following form:

```
[task_limit] = lastname_battery(task_list)
```

- `task_list` is a single vector containing the energy required for each task—this would be a single column from `rover_tasks.xlsx`.
- `task_limit` is the number of tasks that can be completed before 2.5 kW-hrs are expended
- **NOTE: Your function must use a while loop** to determine how many tasks it takes to expend 2.5 kW-hrs. Start with the first task and add up all the energy required by each subsequent task until you have 2.5 kW-hrs.

- c) (20 points) In this problem you must create a scoring function for the Olympic snowboard halfpipe event. This event is scored by 6 judges. To generate a competitor's final score, you must ignore the lowest score and the highest score from the judges and calculate the average of the remaining scores.

Your function call should have the following form:

```
[final_scores] = lastname_snowboard(raw_scores)
```

- `raw_scores` is a matrix of raw scores from the judges (each row is the set of scores for a particular competitor)
- `final_scores` is a single vector containing all the competitors' final scores
- You must ignore the lowest score and the highest score from the judges and calculate the average of the remaining scores

Your main program should:

- Read the Excel file called `raw_scores.xlsx`
- Call the function
- Print the best score from the vector of final scores to the command window:

```
The highest score was 85.5
```

EXAM 2 – WRITTEN PORTION

NAME _____

SECTION NUMBER _____

CAMPUS MAILBOX NUMBER _____

EMAIL ADDRESS _____@rose-hulman.edu

Written Portion	/ 40
Computer Portion	/ 60
Total	/ 100

USE MATLAB SYNTAX FOR ALL PROGRAMS AND COMMANDS YOU WRITE

Problem 1: (4 points) We have loaded a matrix called `pressuredata`. We wish to copy the 7th column of that matrix into a new vector called `tap7`. We do not want to use a for loop. Complete the single line of code below to accomplish this task:

```
tap7=pressuredata(_____);
```

Problem 2: (8 points) We wish to print a table of the cubes of integers for cubes that are less than 100. Complete the code below. (Do not add additional lines—just complete the lines we have provided.)

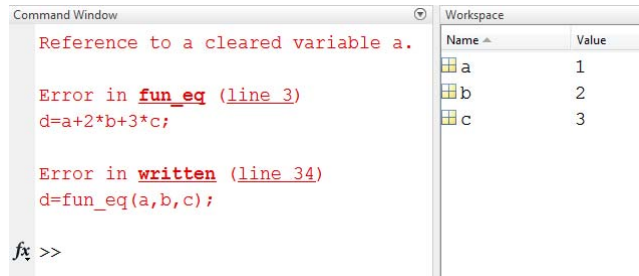
```
clc
clear variables
fprintf('  integer      cube      \n');
integer=1;
cube=1;
while _____
    fprintf('  %2.0f      %2.0f      \n', integer,cube);
    _____
end
```

Problem 3: (4 points) We have created a vector called `Force_x`. Complete this single line of code that will calculate the mean of the entries in `Force_x` from 20 to 45:

```
mean_force_20_45=_____;
```

Problem 4: (4 points) We wrote the main routine and function shown below. When we run the main routine, though, we get an error in the command window. Mark the changes on the code so that it runs correctly.

```
clc
clear variables
a=1;
b=2;
c=3;
d=fun_eq(a,b,c);
fprintf('d is %3.0f \n',d)
```



```
function [d]=fun_eq(a,b,c);
clear variables
d=a+2*b+3*c;
```

Problem 5: (4 points) We wish to create a function to find the distance between two pairs of points, so we have written the following main routine to test our function:

```
clc
clear variables
x1=10;
y1=5;
x2=20;
y2=3;
dist=find_distance(x1,y1,x2,y2);
```

Complete the function below:

```
function [dist]=find_distance(x1,y1,x2,y2);
```

Problem 6: (4 points) Now let us suppose that we have done the previous problem correctly, and we have a distance function that works. We load in two vectors containing the x and y coordinates of 10 points, and we want to find the distances between all the pairs of points. (That is, we want the distance between point 1 and point 2, point 1 and point 3, etc. We will also calculate the distance between point 1 and itself.) Complete the line in the code:

```
clc
clear variables
load x
load y
for i=1:10
    for j=1:10
        dist(i,j)=find_distance(_____);
    end
end
```

Problem 7: (4 points) Consider the main routine and function shown below. When we run the main routine, what will print?

```
clc
clear variables
d=1;
e=2;
f=3;
g=new_fun_eq(d,e,f);
fprintf('g is %3.0f \n',g)

function [d]=new_fun_eq(a,b,c);
d=a+2*b+3*c;
```

- (a) Nothing will print because the program will give an error. You must change the variables in the function to d, e, f, g
- (b) g is 14
- (c) Other. Explain _____

Problem 8: (4 points) Consider the main routine and function shown below. When we run the main routine, what will print?

```
clc
clear variables
a=1;
b=2;
c=3;
d=third_fun_eq(c,b,a);
fprintf('d is %3.0f \n',d)
```

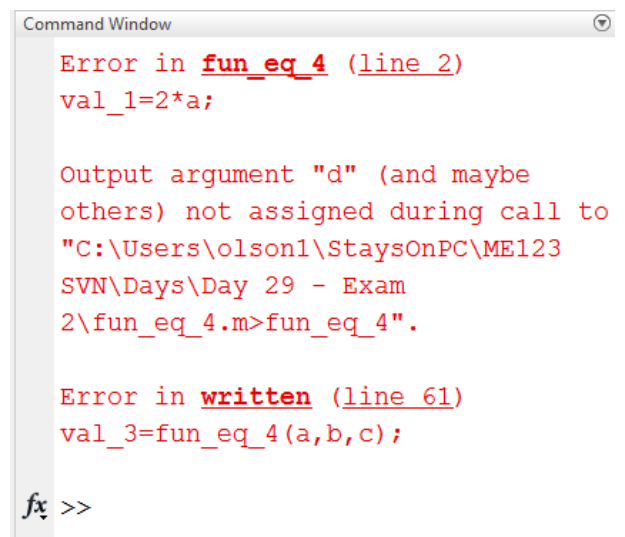
```
function [d]=third_fun_eq(a,b,c);
d=a+2*b+3*c;
```

- (a) Nothing will print because the function won't run. You must change the order of the variables in the function to c, b, a
- (b) d is 14
- (c) d is 10
- (d) Other. Explain _____

Problem 9: (4 points) We wrote the main routine and function shown below. When we run the main routine, though, we get an error in the command window. Mark the changes on the code so that it runs correctly.

```
clc
clear variables
a=1;
b=2;
c=3;
val_3=fun_eq_4(a,b,c);
fprintf('val_3 is %5.2f \n',val_3)
```

```
function [d]=fun_eq_4(a,b,c);
val_1=2*a;
val_2=b*exp(c);
val_3=val_1+val_2;
```



```
Command Window
Error in fun_eq_4 (line 2)
val_1=2*a;

Output argument "d" (and maybe
others) not assigned during call to
"C:\Users\olson1\StaysOnPC\ME123
SVN\Days\Day 29 - Exam
2\fun_eq_4.m>fun_eq_4".

Error in written (line 61)
val_3=fun_eq_4(a,b,c);

fx >>
```