

- 1) (3 pts) Write a MATLAB script that will ask the user to input the coordinates of a point (x, y) in the plane and determine if the point lies in the space between the circles with centers at the origin and radii $r_1 = 2$ and $r_2 = 4$.

```
x = input('Input x: ');
y = input('Input y: ');
d = sqrt(x^2 + y^2);
if(d >= 2 & d <= 4)
    disp('Inside ring')
else
    disp('Not in ring')
end
```

- 2) (4 pts) If you have 4 resistors with resistances R_1, R_2, R_3, R_4 wired in parallel, the equivalent resistance is given by

$$R_{\text{par}} = \frac{1}{\frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{R_4}}.$$

If you have 4 resistors wired in series, the equivalent resistance is given by

$$R_{\text{ser}} = R_1 + R_2 + R_3 + R_4.$$

Write a MATLAB script that will ask the user to input values of R_1, R_2, R_3, R_4 and display only the larger of R_{par} and R_{ser} . Test your script using $R_1 = 5.6, R_2 = 4.1, R_3 = 0.5, R_4 = 9.7$.

```
R1 = input('Input R1: ');
R2 = input('Input R2: ');
R3 = input('Input R3: ');
R4 = input('Input R4: ');
s = 1/R1 + 1/R2 + 1/R3 + 1/R4;
Rpar = 1/s;
Rser = R1 + R2 + R3 + R4;
if(Rpar > Rser)
    disp('Rpar is larger')
    Rpar
else
    disp('Rser is larger')
    Rser
end
```

- 3) (5 pts) Sometimes it is necessary to reverse the positions of the elements of a vector. For example, if

$$x = \begin{pmatrix} 6 \\ 9 \\ 1 \\ 2 \end{pmatrix},$$

you need to change this to

$$x = \begin{pmatrix} 2 \\ 1 \\ 9 \\ 6 \end{pmatrix}.$$

The MATLAB command `flipud` will do this, but you can't use that here. Write a script that will take a vector x of length n and reverse the positions of the elements. The vector x above is just an example of what you need to do. Your script must work for any vector of length n . HINT: Define a temporary vector y the same size as x , create the reversed vector in y then copy y into x .

```

We need
y(n) = x(1)
y(n-1) = x(2)
y(n-2) = x(3)
...
y(1) = x(n)

```

```

-----
y = zeros(size(x));
for i = 1:length(x)
    y(n+1-i) = x(i);
end
x = y;

```

4) (3 pts) Suppose

$$x = \begin{pmatrix} 1 \\ a \\ 3 \end{pmatrix}, \quad y = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix}.$$

Determine that value of a that satisfies the equation $x^T y = 6$.

$$\begin{aligned}
 x^T y &= 1 \cdot 2 + a \cdot 4 + 3 \cdot 3 \\
 &= 4a + 11 \\
 4a + 11 &= 6 \\
 4a &= -5 \\
 a &= -5/4
 \end{aligned}$$

5) (2 pts) Suppose you have the two matrices

$$A = \begin{pmatrix} 1 & 2 & -1 \\ -2 & 3 & 1 \\ 4 & 0 & 3 \end{pmatrix}; \quad B = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

Use MATLAB to verify the identity

$$(A \cdot B)^T = B^T \cdot A^T.$$

```

>> A = [1 2 -1; -2 3 1; 4 0 3];
>> B = [0 1 1; 1 1 0; 1 0 1];
>> (A*B)'
ans =
     1     4     3
     3     1     4
     0    -1     7
>> B'*A'
ans =
     1     4     3
     3     1     4
     0    -1     7

```

6) (3 pts) Explain what physical quantity will be contained in the variable d when the code section below is run.

```

x,y = (some column vectors of equal length that have been defined)
n = length(x);
d = 0;
for i = 1:n
    d = d + sqrt(x(i)^2 + y(i)^2);
end

```

 d will contain the sum of the distances of the (x,y) points from the origin.

7) (2 pt) True or False: The code segment in Question 7 can be accomplished by doing

```
d = sum(sqrt(x^2 + y^2));
```

False; you need `d = sum(sqrt(x.^2 + y.^2));`

8) (6 pts) Recall that the 2-norm of a vector is defined as

$$\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}.$$

The code below is supposed to compute the 2-norm of a vector. Explain all the reasons that this code is incorrect.

```
x = (some vector that has been defined)
n = lenght(x);
for i = 1,n
    total = total + sqrt(x{i^2});
end
```

total should be set to 0 before entering the loop
length is spelled wrong
the for should be: `for i = 1:n`
The {} brackets should be ()
The square is in the wrong place. It should be: `x(i)^2`
the sqrt root is in the wrong place and operating on the wrong quantity.
It should be operating on total outside the main loop.

```
total = 0;
n = length(x);
for i = 1:n
    total = total + x(i)^2;
end
total = sqrt(total);
```