Session 4 Preparation

To be completed in the period between Session 3 and Session 4.

The purpose of this Exercise is to give you practise at program design.

- On the next few pages is a worked example of how to design a program to create String Art.
- Then, the Program Design Challenge page gives a short summary of each design.
- The detailed descriptions of each the design challenge follows.

Work your way through the example, then :-

1 Pick one of the design challenges.

2 Produce a program design.

The easiest why to create your design and program is to create a new directory in MATLAB and use the MATLAB editor to create your design file.

```
>> mkdir MyDesign
>> cd MyDesign
>> edit MyDesign.txt
```

You don't have to call the directory or the file MyDesign.

As an alternative, you can use Notepad or Wordpad in Windows or TextEdit on a MAC.
If you use a word processor such as Word, make sure you save the file as a Plain Text file (.txt), once you have finished.

There is **1 MARK** for submitting a .txt file containing a satisfactory design.

3 Use your program design to build the program

- Copy the design into a MATLAB program.
- Put a % in front of each line so the whole design is a comment in the program.
- Then use the design as a framework for the MATLAB code.

There is **1 MARK** for producing a program using your design.
There is **1 MARK** for a working program.

pto

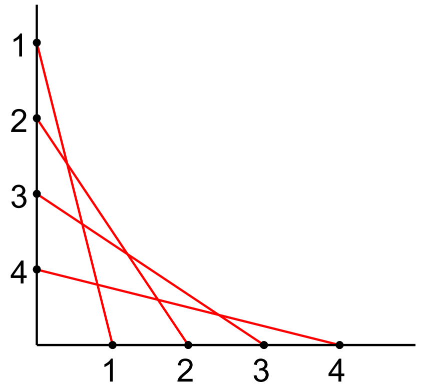
4 Submit you files to Canvas.

Suppose that your **.txt** file containing your design and all the **.m** files for the program are in a folder called **MyDesign**.

You will submit the whole of the **MyDesign** folder as a zip file.

- In MATLAB, go up one level to the folder containing the **MyDesign** folder.
- Right click on the **MyDesign** folder.
- Select **Create Zip File**
  This creates **MyDesign.zip**
- Go to **P5 Engineering Coursework** in Canvas.
  https://canvas.ox.ac.uk/courses/62830
- Click on **Comp4 Prep** under **Computing Laboratory**
- Click on **Submit Assignment**.
- Click on **Choose File** and select the zip file to upload.
- Click on **Submit Assignment.**

Example: String Art



The following program design produces the same sort of pattern as produced in String Art. You will find pictures of String Art on the internet.

Before designing a program, it is useful to think about how you would do the task without a computer. So lets consider how String Art is made.

- You start with a wooded board.
- You nail in a number of tacks, evenly space along a straight line. You leave the heads protruding above the surface for the string.
- You assign the number one to a tack at one end, so that the numbers increase along the line.
- Somewhere else on the board, you nail in a second line of tacks, with the same spacing and the same number of tacks.
- You reverse the numbering so that tack one on this line is farthest from tack one on the original line.
- Then join tack one on the origin line to tack one on the other line with thin, brightly coloured string.
- Then you take the string to the next tack in the line and join tack two to tack two.
- Repeat for each pair of tacks.

In the program, we will use points on a graph instead of tacks. To keep things simple, we will use a line of points on the x axis and a second line of points on the y axis. Instead of string, we join the points with a straight line.

When writing a document, you may start by writing the headings and then add the detail later. That is exactly how you design a program. At the top level of the program design, you write out the headings, without the detail. At this stage, you want a minimal number of steps.

Demonstrate String Art - Design 1

Producing the kind of patterns produced by String Art on a computer.

```
1 Define the points on the x axis.
2 Define the points on the y axis, in reverse to the x axis
3 Find N, the number of points on the x axis
4 Repeat for each line k for k=1 to k=N
5       Draw a line from the $k^{th}$ point on the x axis to the $k^{th}$ point on the y axis.
6 End Repeat
```

Then you can add more detail.

Demonstrate String Art - Design 2

Producing the kind of patterns produced by String Art on a computer.

```
1  Define the points on the x axis.
    1.1 Each point is 0.01 apart, starting at 0.01 and ending at one.
2  Define the points on the y axis, in reverse to the x axis
    2.1 Each point is 0.01 apart, starting at one and ending at 0.01.
3  Find N, the number of points on the x axis
4  Repeat for each line k for k=1 to k=N
5       Draw a line from the $k^{th}$ point on the x axis to the $k^{th}$ point on the y axis.
        5.1  Vector Vx is the x coordinates of the $k^{th}$ line.
        5.2  Vector Vy is the y coordinates of the $k^{th}$ line.
        5.3   Plot Vx , Vy
        5.4  Hold the plot on the graph
6  End Repeat
```

You should by now have a good idea of how to convert this design into a program. To make it clearer what you need to do, I am going to add a few more details about the coordinates.

Demonstrate String Art - Design 3

Producing the kind of patterns produced by String Art on a computer.

```
1  Define the points on the x axis.
   1.1  Each point is 0.01 apart, starting at 0.01 and ending at one.
2  Define the points on the y axis, in reverse to the x axis
   2.1  Each point is 0.01 apart, starting at one and ending at 0.01.
3  Find N, the number of points on the x axis
4  Repeat for each line k for k=1 to k=N
5        Draw a line from the $k^{th}$ point on the x axis to the $k^{th}$ point on the y axis.
         5.1  Vector Vx is the x coordinates of the $k^{th}$ line.
              5.1.1  The 1st x coordinate will be the $k^{th}$ point on the x axis.
              5.1.2  The 2nd x coordinate is zero because it is on the y axis.
              5.1.3  Put both x coordinates into the vector Vx
         5.2  Vector Vy is the y coordinates of the $k^{th}$ line.
              5.2.1  The 1st y coordinate is zero because it is on the x axis.
              5.2.2  The 2nd y coordinate will be the $k^{th}$ point on the y axis.
              5.2.3  Put both y coordinates into the vector Vy
         5.3  Plot Vx , Vy
         5.4  Hold the plot on the graph
6  End Repeat
```

- Then you copy the design into a MATLAB program.
- Put a % in front of each line so the whole design is a comment in the program.
- Then you use the design as a framework for the MATLAB code.

I suggest that you implement the program design above into MATLAB code. Then compare your program with my solution on page 11.

In this example, I have attempted to produce a program that not only works, but is easily understood by any Engineering undergraduate. I hope that you found that it is very clear what the program does and how it does it. When you produce your own designs, you need to aim for the same goal. Assume that your program is going to be read by other students. Then think about how easy it is to understand. You should be proud of your program and be willing to let other people read it.

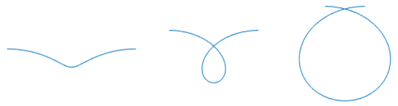
Program Design Challenges

These exercises have been developed to give you more experience creating larger MATLAB programs, using a Top Down Design approach. Below is a summary of each of the problems. Try the problem that sound interesting to you!

Design 1: Morphing

A script with ~50 lines and a short function

Morphing is the process to slowly turn one shape into another shape. Create a script that can animate turning a straight line into a circle.



Design 2: Cipher

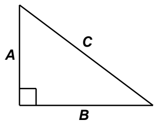
One short test script and three functions
3 functions each with ~30 lines

A cipher is a type of code used to keep messages secret. Each letter of the alphabet is replaced by a symbol such as a number. Create functions to generate ciphers, encode and decode messages.

Design 3: Pythagorean Triple

Code one script ~30 lines

A Pythagorean Triple is a set of three integers ***A***, ***B*** and ***C*** such that $A^2 + B^2 = C^2$ The classic example of a Pythagorean Triple is the 3, 4, 5 triangle. Design a script to print out all the Pythagorean Triples where ***A*** and ***B*** are less than a specified number ***n***.



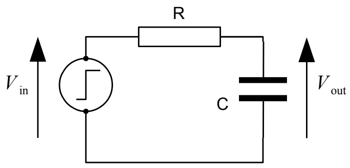
Design 4: Resistor and Capacitor Circuit

One script ~50 lines

At time ***t***, the output voltage V_{out} of a given circuit is given by:

$$V_{out} = V_{in}\left(1 - \exp\left(-\frac{t}{RC}\right)\right)$$

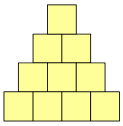
For a given V_{in} and ***t*** the value of V_{out} will change depending on the resistor ***R*** and capacitor ***C***. Design a program to find the best capacitor and resistor, out of a list of available components, that will give the value of V_{out} closest to a specified value.



Design 5: Pyramid

A ~30 line script and some short functions

Design a program to draw a pyramid of squares as shown on the right, given a specific number of squares on the base of the pyramid.



Design 1: Morphing

Morphing is the process of using a computer to slowly turn one shape into another shape.

Suppose you have two shapes with $\boldsymbol{n}$ coordinates each. So

$\boldsymbol{X}_1$ and $\boldsymbol{Y}_1$ are vectors containing the coordinates of shape 1.

$\boldsymbol{X}_2$ and $\boldsymbol{Y}_2$ are vectors containing the coordinates of shape 2.

Then

$$\boldsymbol{X} = (\mathbf{1} - \alpha)\boldsymbol{X}_1 + \alpha \boldsymbol{X}_2 \qquad \mathbf{0} \le \alpha \le \mathbf{1}$$
$$\boldsymbol{Y} = (\mathbf{1} - \alpha)\boldsymbol{Y}_1 + \alpha \boldsymbol{Y}_2$$

defines a new shape that is part way between the two shapes.

When $\alpha = \mathbf{0.25}$ the new shape is three quarters shape 1 and a quarter shape 2.

When $\alpha = \mathbf{0.5}$ the new shape is a half shape 1 and a half shape 2.

To slowly change shape 1 into shape 2, you animate the process so that $\boldsymbol{\alpha}$ slowly changes from zero to one.

Design a script that morphs a straight line, between -1 and 1 on the y axis to a circle of radius one, centred on the origin.

Hint
To see how to do animation, look back to your solution of Exercise 6 in Session 2.

Design 2: Cipher

A cipher is a type of code used to keep messages secret. The following is a cipher key.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
p	n	a	.	j	k	y	x	l	s	c	t	i	d

15	16	17	18	19	20	21	22	23	24	25	26	27	28
z		e	q	r	f	o	w	b	u	g	m	h	v

Each character is associated with a number that can be used to represent that character in an encrypted message. The table above can be used to convert a message into code.

where are you → 22 27 17 19 17 16 3 19 17 16 7 21 24

and to convert a coded message back into text.

21 2 16 12 27 17 16 12 19 3 13 2 → **on the train**

A MATLAB character array can be used as a cipher key. The character array below represents the cipher key above.

Key = 'pna.jkyxlsctidz eqrfowbugmhv'

Key(6) is the sixth character of Key, the letter **k** in the above key.

(a) Design a function called **CipherGen** to create a key for a cipher.
key = CipherGen()

You should get a different cipher key each time you run the function.

Hints
Start with an ordered list of characters.
Try the following in the MATLAB command window.

```
>>  'a' : 'z'
>> [  'a' : 'z'  '. ']
```

← full stop then a space

Then mix it up by swapping characters around.

(b) Design a function that uses the cipher key to encipher a text message into a coded message.
code = encipher(key,message)

Hint
Find out what the **strcmp** and **stccmpi** functions do. It is possible that you will find them useful.

(c) Design a function that uses the cipher key to decipher a coded message into a readable text.
message = decipher(key,code)

(d) The following phrase contains all the letters of the alphabet:

the quick brown fox jumped over the lazy dog

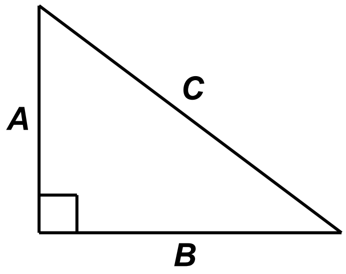
Use your function to encipher the phrase into code and then see if you can decipher the code with the other function.

Design 3: Pythagorean Triple

A Pythagorean Triple is a set of three integers ***A***, ***B*** and ***C***, such that

$$A^2 + B^2 = C^2$$

So that ***A***, ***B*** and ***C*** represent the lengths of the three sides of a right angle triangle. The classic example of a Pythagorean Triple is the 3, 4, 5 triangle.



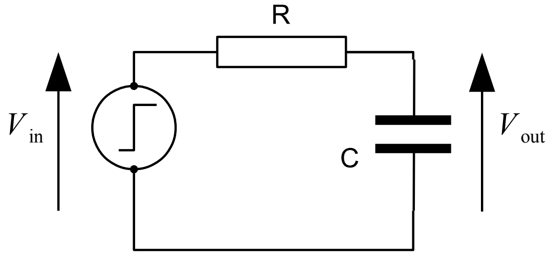
Design a script to print out all the Pythagorean Triples where ***A*** and ***B*** are less than a specified number ***n***.

Hint
The following example should help you to determine if ***C*** is an integer.

```
>>  C = sqrt(25)
>>  rem(C,1)
>>  C = sqrt(26)
>>  rem(C,1)
```

Design 4: Resistor and Capacitor Circuit

At time t , the output voltage V_{out} of the circuit on the right is given by:



$$V_{\text{out}} = V_{\text{in}}\left(1 - \exp\left(-\frac{t}{RC}\right)\right)$$

Only certain values of resistors and capacitors are available.

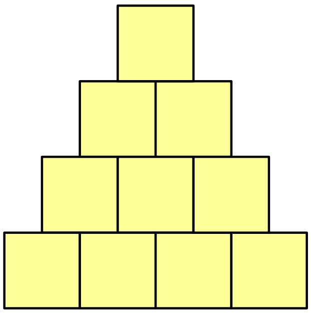
The Capacitor Values Available										1 nF = 10^{-9} Farads			
10	12	15	18	22	27	33	39	47	56	68	82	100	nF

The Resistor Values Available											1 KΩ = 10^3 Ω		
6.8	7.5	8.2	9.1	10	11	12	13	15	16	18	20	22	KΩ
24	27	30	33	36	39	43	47	51	56	62	75	82	

The input voltage V_{in} is 4 volts. It is required that the output voltage V_{out} reaches 3 volts when t = 1 millisecond.

Design a program to find the value of V_{out} with the smallest error.
Then list all the values of **R** and **C** that will give the smallest error.

Design 5: Pyramid



Design a program to draw a pyramid of squares as shown on the right.

Let ***n*** define the number of squares at the base of the pyramid. Design the program so that ***n*** can be changed easily, so that the program can draw pyramids of different size.

Hint: drawshape.m and **translate.m** from the Exercises
in Session 2 could help you here.

StringArt.m

```
% Demonstrate String Art
% Producing the kind of patterns produced by String Art on a computer.


% 1 Define the points on the x axis.
% 1.1 Each point is 0.01 apart, starting at 0.01 and ending at one.
X = 0.01:0.01:1;

% 2 Define the points on the y axis
% 2.1 Each point is 0.01 apart, starting at one and ending at 0.01.
Y = 1:-0.01:0.01;

% 3 Find N, the number of points on the x axis
N = length(X);

% 4 Repeat for each line k for k=1 to k=N
for k = 1:N
   % 5 Draw a line from kth point on x axis to kth point on y axis.

   % 5.1 Vector Vx is the x coordinates of the line.
   % 5.1.1 The 1st x coordinate will be the kth point on the x axis.
   x1 = X(k);

   % 5.1.2 The 2nd x coordinate is zero because it is on the y axis.
   x2 = 0;

   % 5.1.3 Put both x coordinates into the vector Vx
   Vx = [ x1 x2 ];

   % 5.2 Vector Vy is the y coordinates of the kth line.
   % 5.2.1 The 1st y coordinate is zero because it is on the x axis.
   y1 = 0;

   % 5.2.2 The 2nd y coordinate will be the kth point on the y axis.
   y2 = Y(k);

   % 5.2.3 Put both y coordinates into the vector Vy
   Vy = [ y1 y2];

   % 5.3 Plot Vx , Vy
   plot(Vx,Vy,'r');

   % 5.4 Hold the plot on the graph
   hold on

end    %6  End Repeat
```