Possible statements/syntax in the `MA2895` class test

- Creating vectors and matrices, e.g. `[` and `]`, a comma to separate entries on a row, a semi-colon to separate rows, the use of the transpose `'`. Combining matrices to create larger matrices.
- `*` and `^` as matrix operations.
- Entry-wise operations such as `.*`, `.^` and `./` etc.
- The use of `&&` (logical and) and `||` (logical or).
- The use of the colon notation to extract parts of vectors and matrices.
- Decision statements, e.g. `if` and `if–else` constructions.
- `for`–loops
- `break` and `continue` in a loop.
- Basic use of `fprintf` for formatted output.
- The `function` statement at the top of `function files`.

Entry-wise operations

`.*`, `./` and `.^` are entry-wise operations. An example which also uses standard functions in a vectorised way is as follows.

Suppose we want to plot

$$f(t) = \sin(t) + 0.3\exp(-0.1t^2)\sin(10t).$$

```
t=linspace(0, 2*pi, 500);
y=sin(t)+0.3*exp(-0.1*t.^2).*sin(10*t);
figure(2)
plot(t, y)
```

The one statement involving `y=` achieves what the following 4 statements do in creating the vector `z`.

```
z=zeros(1, 500);
for k=1:500
  z(k)=sin(t(k))+0.3*exp(-0.1*t(k)^2)*sin(10*t(k));
end
```

Matrix operation/entry-wise operation comparison

```
A=[4  1  1;
   1  4  1;
   1  1  4];
A2=A*A
E2=A.*A
```

This was also in the week 18 handout and it creates the following.

```
A2 =
    18     9     9
     9    18     9
     9     9    18
E2 =
    16     1     1
     1    16     1
     1     1    16
```

Function file example ..mathematical specification

Suppose you wish to evaluate the finite Fourier series

$$g_m(x) = \frac{4}{\pi}\left(\sin(x) + \frac{\sin(3x)}{3} + \frac{\sin(5x)}{5} + \cdots + \frac{\sin((2m+1)x)}{2m+1}\right).$$

The function depends on x and m. Hence to mimic this in Matlab we want a function with these as the two input parameters and we just want to create one output. We can do this in a vectorised way with little additional effort. The structure of a file called `g.m` can be as follows.

```
function y=g(x, m)
% ... statements to set y
```

In the function we will need a loop.

Function file example ..the Matlab part

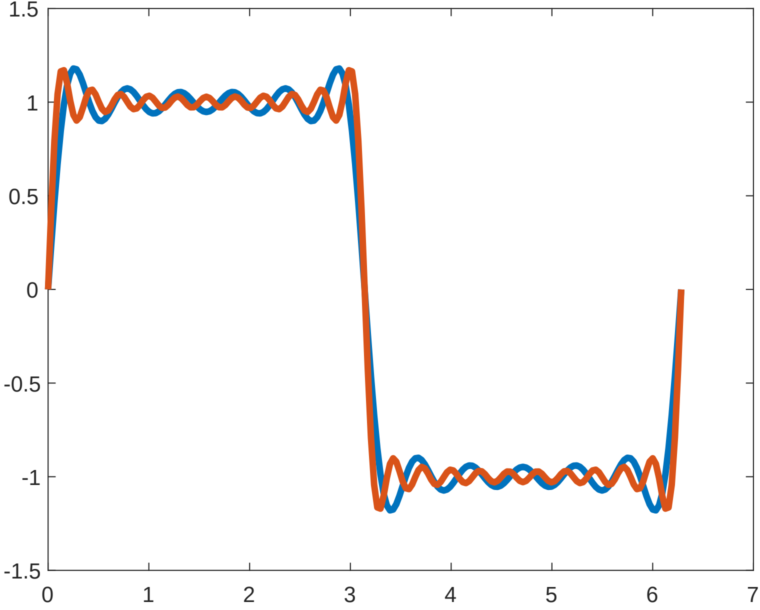
```
function y=g(x, m)

y=zeros( size(x) );
for k=0:m
  y=y+sin( (2*k+1)*x )/(2*k+1);
end
y=4*y/pi;
```

We can use this elsewhere to plot g_5 and g_{10} with statements such as the following.

```
x=linspace(0, 2*pi, 201);
figure(20)
plot(x, g(x, 5), x, g(x, 10), 'LineWidth', 3);
```

Plot of $g_5(x)$ and $g_{10}(x)$



An example of counting – an `if–end` block

From the previous plot most of the curve seems to be close to 1 or -1. You can crudely quantify this by counting how many values are outside of the interval $(-0.9, 0.9)$.

```
n=2001;
x=linspace(0, 2*pi, n);
y=g(x, 10);
count=0;
for i=1:n
  if y(i)<=-0.9 || y(i)>=0.9
    count=count+1;
  end
end
fprintf('With %d evaluations, %d are outside\n', ...
        n, count);
```

Other ways of doing the counting

We could use `abs()` to shorten the test statement.

```
count=0;
for i=1:n
  if abs(y(i))>=0.9
    count=count+1;
  end
end
```

Another possibility is to skip the ones we do not want to count.

```
count=0;
for i=1:n
  if abs(y(i))<0.9
    continue;
  end
  count=count+1;
end
```