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.

Matrices ...what is displayed?

```
x=[7  8  2  4]';
y=[1  1  1  2]';

A=[0  2  4  6  8;
   7  5  3  1  0;
   9  9  4  1  4;
   7  8  8  7  8;
   5  4  3  2  1];

v1=A(4, 5)
v2=x+A(1:4, 1)

v3=A(1:2:5, 5)
v4=A(end, 2:4)

v5=A([3 3 3], 1:3)
v6=[x  y  y  x]
```

Loops ... what is displayed?

A `for`–loops and a `break` statement

```
for k=1:4
  y=k^3+k^2;
  fprintf('k=%d, y=%2d\n', k, y);
end
```

```
for k=1:4
  y=k^3+k^2
  if y>=30
    break
  end
end
k
```

Loops ... what is displayed?
for–loops, break and continue statements

```
for k=1:5

  y=k^3+k^2;

  if y<=20
    continue;
  end

  if y>100
    break;
  end

  disp(y)
end
k
```

Matrix operations ...what is displayed
Matrix multiplication, entry-wise operations

```
A=[1  2;
   5  4;
   7  6];

B=A.^2

C=A'*A

E=A*A'
```

Creating a function

Let $\underline{x} = (x_i)$ be a vector of length $m \geq n$ and let

$$f_n(\underline{x}) = \begin{cases} \|\underline{x}\|_1, & \text{if } 1 \leq n \leq 3, \\ x_4^2 + \cdots + x_m^2 + \|\underline{x}\|_1, & \text{otherwise.} \end{cases}$$

Write a function starting with the following which computes this.

```
function y=fun20(x, n)
m=length(x);
```

Which function computes ···?

Suppose we want a function to compute

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

Which of the following functions works correctly for a scalar x and for all $m \geq 0$.

```
function y=gm1(x, m)
k=1:2:(2*m+1);
y=sin(k*x)*(ones(m+1, 1)./k(:));
```

```
function y=gm2(x, m)
y=0;
for k=2*m+1:2:1
  y=y+sin(k*x)/k;
end
```

Which function computes ⋯ continued?

```
function y=gm3(x, m)
y=0;
s=0;
for k=0:m
  s=s+sin((2*k+1)*x)/(2*k+1);
end
```

```
function y=gm4(x, m)
y=0;
for k=1:m
  y=y+sin((2*k+1)*x)/(2*k+1);
end
```

Slide 2 output, `v1` to `v4`

The output from slide 2 statements is as follows.

```
v1 =
     8
v2 =
     7
    15
    11
    11
v3 =
     8
     4
     1
v4 =
     4     3     2
```

Slide 2 output, `v5` to `v6`

```
v5 =
     9     9     4
     9     9     4
     9     9     4
v6 =
     7     1     1     7
     8     1     1     8
     2     1     1     2
     4     2     2     4
```

Note that the part `[3 3 3]` means that a part of row 3 is repeated when creating `v5`.

Slide 3 output

In the first case it is as follows.

```
k=1, y= 2
k=2, y=12
k=3, y=36
k=4, y=80
```

Slide 3 output continued

In the second case it is as follows.

```
y =
     2
y =
    12
y =
    36
k =
     3
```

The `break` statement means that the loop finishes when `k=3` and as `k` is displayed after the loop this is the last value shown.

Slide 4 output

The output is as follows.

```
    36
    80
k =
     5
```

When `k=1` and when `k=2` the value of `y` is less than 20 and the `continue` statement is executed and thus there is no output.

When `k=3` and when `k=4` both tests are false and and the statement where `y` is displayed is reached which is why 36 and 80 are shown.

When `k=5` the test attached to the `break` statement is true and we leave the loop. After the loop `k` still contains the value 5 and this is the last output.

Slide 5 output

The output is as follows.

```
B =
    1     4
   25    16
   49    36
C =


   75    64
   64    56
E =
    5    13    19
   13    41    59
   19    59    85
```

Remember that it is an entry-wise operation to get `B` but for `C` and `E` symmetric matrices are created and `*` means matrix multiplication.

Slide 6 – a possible version of the function

One possibility for the function which follows the description quite closely is the following.

```
function y=fun20(x, n)
m=length(x);

if 1<=n && n<=3
  y=norm(x, 1);
else
  y=norm(x, 1);
  for k=4:m
    y=y+x(k)^2;
  end
end
```

Slide 6 – another version of the function

A shorter version exploiting Matlab capabilities and avoiding the loop is to have the following.

```
function y=fun20b(x, n)
m=length(x);

y=norm(x, 1);
if 1<=n && n<=3
  return
end
y=y+sum( x(4:m).^2 );
```

Slide 7 – which functions work correctly

`gm1.m` is correct. `k` is a row vector and thus `sin(k*x)` is a row vector which deals with computing all the sine terms. The later part creates a column vector of the "`1/k`" terms and the row vector times the column vector gives the required sum.

`gm2.m` is not correct. The part `2*m+1:2:1` gives an empty vector and thus `y` remains at `0`. The step needs to change to `-2` to work.

Slide 8 – which functions . . . correctly

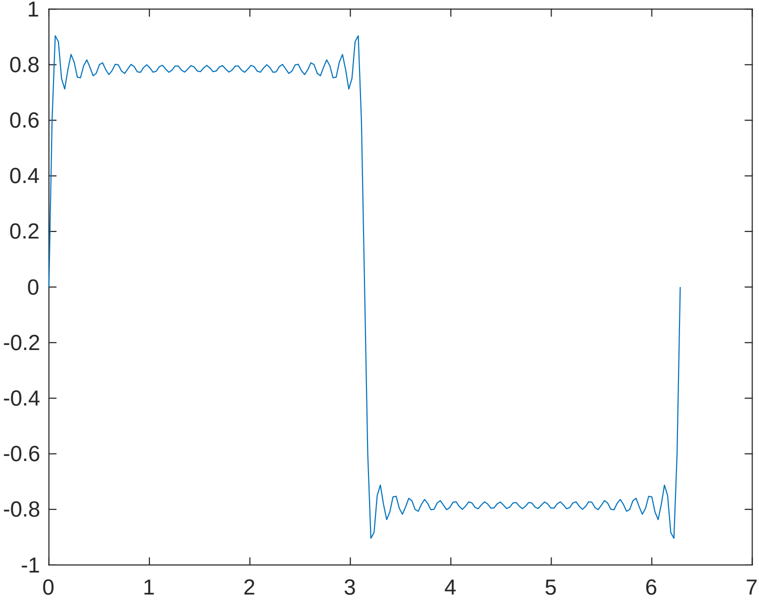
`gm3.m` is not correct as `y` is set to 0 and it is never changed. The sum is computed locally as `s` but this is not returned by the function.

`gm4.m` is not correct as the term `sin(x)` when `k` is 0 is not included.

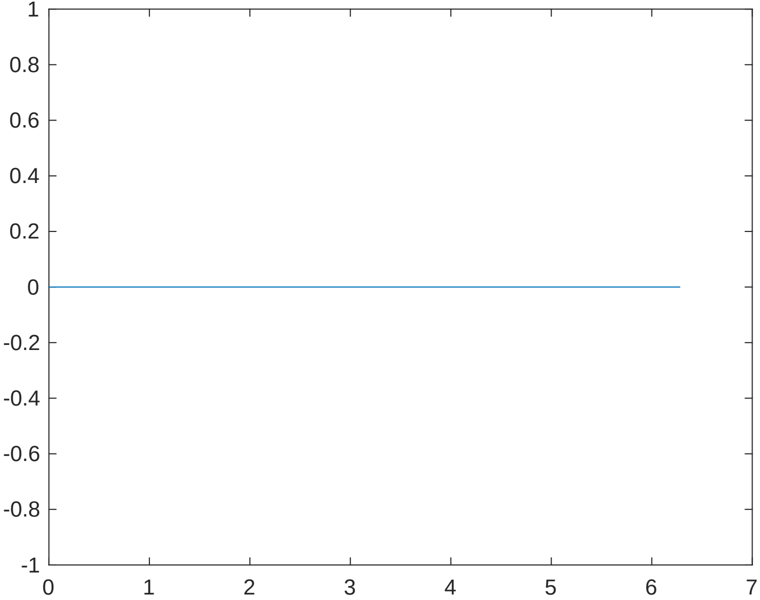
Some tests with gm1, gm2, gm3 and gm4

```
x=linspace(0, 2*pi, 201);
y1=zeros(1, 201);
y2=y1;  y3=y1;  y4=y1;
for k=1:201
  y1(k)=gm1(x(k), 20);
  y2(k)=gm2(x(k), 20);
  y3(k)=gm3(x(k), 20);
  y4(k)=gm4(x(k), 20);
end
figure(101)
plot(x, y1);   print2pdf('gm1plot.pdf')
figure(102)
plot(x, y2);   print2pdf('gm2plot.pdf')
figure(103)
plot(x, y3);   print2pdf('gm3plot.pdf')
figure(104)
plot(x, y4);   print2pdf('gm4plot.pdf')
```

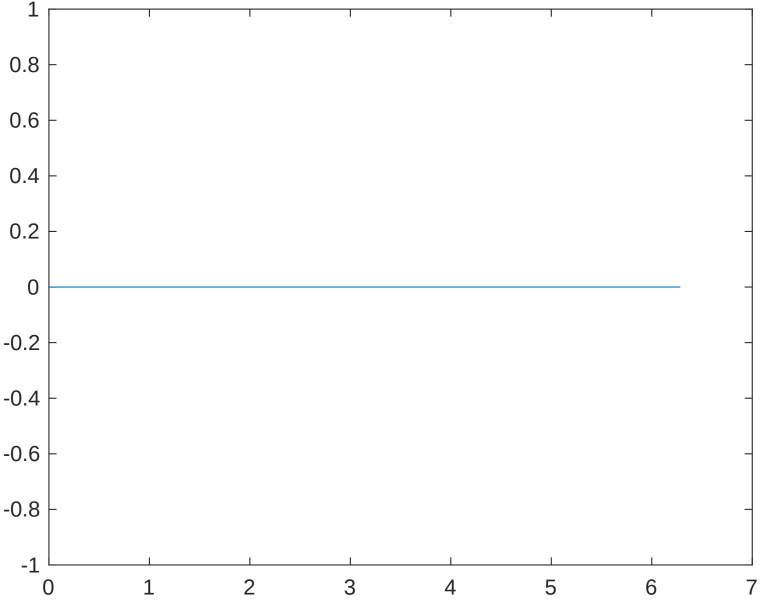
gm1plot.pdf



gm2plot.pdf



gm3plot.pdf



gm4plot.pdf