Problem 1.
The following MATLAB fprintf statements are executed. Give the output appropriately indented. Do not forget to also place the MATLAB cursor! You write down your output in the space after the printf but before the comment line; the numbered comment lines indicate the edge of the screen, with the percent sign indicating not only a MATLAB comment but also column/character position one.

```
>> fprintf('MATLaB%%''\\');

%234567890123456789012345678901234567890123456789012345678901234567890

>> fprintf('MATLaB%10sMATLAB%%\n','matlab');

%234567890123456789012345678901234567890123456789012345678901234567890

>> fprintf('A\nB\nC\n%s','matlab');

%234567890123456789012345678901234567890123456789012345678901234567890

>> fprintf('A\nB\nC\n%10s','matlab');

%234567890123456789012345678901234567890123456789012345678901234567890

>> fprintf('a= %10d b= %5d c= %10d  d=%+5d\n',1000,2000,3000,400);

%234567890123456789012345678901234567890123456789012345678901234567890

>> fprintf('%-10.3f %+10.3f %10.3f %10.3f\n',1111.11,2222.222,3333.333,444.4);

%234567890123456789012345678901234567890123456789012345678901234567890
```

Problem 2.
What does the following fprintf statement print?

```
>> fprintf('%-10.3f %+5.3f %8.3f %5.3f\n',1111.11,2222.222,3333.333,444.4);

%234567890123456789012345678901234567890123456789012345678901234567890
>> fprintf('%-9.3f\n%9.3f\n%+9.3f\n%9.2f\n',1234.567,1234.567,1234.567,1234.567);

%234567890123456789012345678901234567890123456789012345678901234567890
```

Problem 3.
What does the following MATLAB program print
(a) if `var1 = 1`, `var2 = 0`, `var3 = 0`,
(b) if `var1 = 0`, `var2 = 1`, `var3 = 1`,
(c) if `var1 = 1`, `var2 = 1`, `var3 = 0`,
(d) if `var1 = 0`, `var2 = 0`, `var3 = 1`,
(e) if `var1 = 0`, `var2 = 0`, `var3 = 0`.

```
  if (var1)
    fprintf('Block 1\n');
  elseif (var2)
    fprintf('Block 2\n');
  elseif (var3)
    fprintf('Block 3\n');
  else
    fprintf('Block 4\n');
  end
```

Problem 4.
For an input consisting of keys a, b, c what does the code below do? What does it display? Explain. Can you rewrite it so that it becomes less confusing (and simpler)?

```
if (a < b)
    if (a < c)
        disp(a);
    else
        disp(c);
    end
else
    if (b < c)
        disp(b);
    else
        disp(c);
    end
end
```

Problem 5.
We would like to flip a coin randomly and fairly and report `H` (for heads) or `T` (for tails). Use the `rand` function to do that.

Problem 6.
You are given the angles a, b, c of a triangle. Complete the if-then-else statement(s) below, to properly classify it as equilateral (all three equal to 60 degrees), scalene (no side equal), or isosceles (only two sides equal)

```
a=input('a ');
b=input('b ');
c=180-a-b;   % this guarantees that a+b+c is equal to 180 degrees!! no need to check for triangleness!!
fprintf('a=%3d b=%3d  c=%3d  a+b+c=%3d\n',a,b,c,a+b+c);

if (            )
  disp('equilateral');
else
  if (                  )
    disp('isosceles');
  else
    disp('scalene');
  end
end
fprintf('\n');
```

Turn page for solutions

Problem 1.

```
%234567890123456789012345678901234567890123456789012345678901234567890
>> fprintf('MATLaB%%''\\');
MATLaB%'\>>

%234567890123456789012345678901234567890123456789012345678901234567890
>> fprintf('MATLaB%10sMATLAB%%\n','matlab');
MATLaB    matlabMATLAB%
>>

%234567890123456789012345678901234567890123456789012345678901234567890
>> fprintf('A\nB\nC\n%s','matlab');
A
B
C
matlab>>

%234567890123456789012345678901234567890123456789012345678901234567890
>> fprintf('A\nB\nC\n%10s','matlab');
A
B
C
    matlab>>

%234567890123456789012345678901234567890123456789012345678901234567890
>> fprintf('a= %10d b= %5d c= %10d  d=%+5d\n',1000,2000,3000,400);
a=       1000 b=  2000 c=       3000  d= +400
>>
%234567890123456789012345678901234567890123456789012345678901234567890
>>  fprintf('%-10.3f %+10.3f %10.3f %10.3f\n',1111.11,2222.222,3333.333,444.4);
1111.110    +2222.222    333.333    444.400
>>
```

Problem 2.

```
>> fprintf('%-10.3f %+5.3f %8.3f %5.3f\n',1111.11,2222.222,3333.333,444.4);
1111.110   +2222.222 3333.333 444.400
>>
>> fprintf('%-9.3f\n%9.3f\n%+9.3f\n%9.2f\n',1234.567,1234.567,1234.567,1234.567);
1234.567
 1234.567
+1234.567
  1234.57
```

Problem 3.

(a) Block 1
(b) Block 2
(c) Block 1
(d) Block 3
(e) Block 4

Problem 4.

The code determines the smallest of three numbers a, b, c. If $a < b$, the minimum is one of a, c but definitely not b; lines 2-5 compute and display one or the other. If $a > b$, similarly, a can not be the minimum. Of one b,c is. Note that finding the minimum of three keys requires two comparisons. The first is $a < b$. The second is one of $a < c$ (if the first comparison returned true) and $b < c$ (if the first returned false).

A code fragment that does the same but is more elegant (or less confusing) is the following one.

```
output=a;                              % a is the minimum unless
if (b < output); output = b; end       % b is smaller than a
if (c < output); output = c; end       % c is smaller than both
disp(output);
```

Problem 5.

Function `rand` returns a random number in the interval $(0, 1)$. Split the interval in half. The lowest half is an `H` and the upper half is a `T` for r which is the value returned by `rand(1)`.

```
r=rand(1);
if (r < 0.5)
  fprintf('H\n');
else
  fprintf('T\n');
end
```

Problem 6.

```
a=input('a ');
b=input('b ');
c=180-a-b;
fprintf('a=%3d b=%3d  c=%3d  a+b+c=%3d\n',a,b,c,a+b+c);

if ( a==b & b==c)
  disp('equilateral');
else
  if (a==b | b==c | a==c)
    disp('isosceles');
  else
    disp('scalene');
  end
end
fprintf('\n');
```