Problem 1.

What does the following switch/case statement print?

```
A= [  1  3  4  1 7 4 8  ];                    %
n= length(A);                                 %
for k = 1:n                                   %
 switch mod(A(k),3)                           %
   case 0                                     %
    fprintf(' yes');                          %
   case 1                                     %
    fprintf(' no ');                          %
   otherwise                                  %
    fprintf(' maybe\n');                      %
  end                                         %
 end                                          %
```

Problem 2.

What does the following while loop do? Explain.

```
%A= ceil(10*rand(1,20));
n=length(A);
maxindex=1;
maxval  =A(1);
ii=2;
while (ii < n)
  if A(ii) > maxval
    maxval=A(ii);
    maxindex=ii;
  end
  ii=ii+1;
end
maxval, maxindex
```

Problem 3.
Consider the following MATLAB functions `ps7p1` and `ps7p1run` defined in files `ps7p1.m` and `ps7p1run.m` respectively and shown below one after the other.
If we invoke function `ps7p1run` in the MATLAB command window what is going to be printed on screen? Show all the information in the order output.

```
function z=ps7p1(x,y)
   a= 0;
   b= 0;
   y= y+ b;
   x= b+ x;
   b= x+ b;
   a= y;
   z=10;
   fprintf('a=%2d b=%2d \n',a,b);
end

function ps7p1run
   a= 5 ; b= 7;
   c=ps7p1(a,b);
   fprintf('a=%2d b=%2d \n',a,b);
   fprintf('c=%2d \n',c);
end
```

Problem 4.
What does the following MATLAB function `ps7p7` do?

```
  function count=ps7p7(a,b)
    count=0;
    for ii=a:b
       if (mod(ii,2) == 1)
            count = count+1;
       end
    end
  end
```

Problem 5.
MATLAB variable `aaaa` and `bbbb` contain two values respectively. We want to swap the values i.e. `aaaa`'s will become `bbbb`'s value and the other way around. Write a brief sequence of MATLAB commands that can achieve this `swap`.

Problem 6.
(a) Name 6 types of functions in MATLAB.
(b) Name 3 types of variables in MATLAB.

Problem 7.
What does function `ps7p5` print when it get invoked from the command line?

```
function ps7p5
  function ps7my5
   fprintf('Nested\n');
  end
  ps7my5;
end

function ps7my5
  fprintf('Subfunction\n');
end
```

Problem 8.
What does function `ps7p5` print when it get invoked from the command line? The addition to that problem is the existence of a primary function `ps7my5` defined in file `ps7my5.m`

```
function ps7p5           % In file ps7p5.m these three functions are defined

  function ps7my5
   fprintf('Nested\n');
  end
  ps7my5;
end

function ps7my5
  fprintf('Subfunction\n');
end


% Function ps7my5 below is defined in a separate file ps7my5.m

function ps7my5
 fprintf('Primary function\n');
end
```

Turn page for solutions

Representative Solutions to the listed problems

Problem 1.

It prints **no yes no no no no maybe** as in (note the spaces); the dashes over the output were intentionally inserted during my typing of the solutions to highlight the spacing of the output in the following line.

```
-----------------------------
 no  yes no  no  no  no  maybe
```

Problem 2.

The algorithm for the maximum is going to be incremental.

In the `ii`-th iteration it will find the maximum of the first ii keys of `A(1), A(2), ... , A(ii)`. The maximum will be stored in maxval, and the index of A that stores that maximum will be stored in maxindex. After the `ii=n`-th ieration maxval will contain the maximum of the n keys and maxindex the index of A that stores that value (maximum).

We start the algorithm for the `ii=1` iteration by finding the maximum of `A(1), ... , A(1)` that is the maximum of one key `A(1)`. The maximum of one key is the key itself. This is A(1) and this is effected in lines 2 (for the index) and 3 (for the value). This is the base case, i.e. the boundary case of the incremental algorithm.

For the next iteration we increment ii by one i.e. its value will be set to two (line 4). The intent would be to find the maximum of `A(1), ... , A(2)`. After we finish with a given iteration we increment ii by one through the statement of line 10. In general at the end of iteration `ii-1` the maximum of `A(1), A(2), ..., A(ii-1)` will be stored in `maxval`, and in the following iteration `ii` the maximum of `A(1), A(2), ..., A(ii-1) A(i)` will be computed. For this we need lines 6-9. Currently before the start of the ii-th iteration, maxval stored the maximum of the `A(1), A(2), ..., A(ii-1)` keys. To determine whether A(ii) is larger (and thus it is the maximum of the first ii keys or not) we compare A(ii) to maxval in line 6. If indeed A(ii) is larger than maxval, then maxval is set to A(ii)

```
%A= ceil(10*rand(1,20));
n=length(A);                % Line 1
maxindex=1;
maxval  =A(1);              % Line 3
ii=2;
while (ii < n)              % Line 5
  if A(ii) > maxval
    maxval=A(ii);           % Line 7
    maxindex=ii;
  end                       % Line 9
  ii=ii+1;
end                         % Line 11
maxval, maxindex            % Line 12
```

Problem 3.

The output would look like (the first line invokes `ps7p1run`. Roughly, the two argument ps `ps7p1` get swapped.

```
>> ps7p1run
a= 7 b= 5
a= 5 b= 7
c=10
```

Problem 4.

It counts the number of odd numbers between a and b inclusive of the end points.

Problem 5.

We need one extra variable to accommodate the swap. Let it be `temp`

```
>> aaaa;
>> bbbb;
>> temp= aaaa;     % store aaaa into temp
>> aaaa= bbbb;     % overwrite aaaa by bbbb (we have already backedup aaaa into temp).
>> bbbb= temp;     % store in  bbbb the backed-up value of aaaa stored in temp
```

Problem 6.

(a) Six types of functions: (primary/regular) functions, nested functions, subfunctions, function functions, private functions, and anonymous functions.

(b) Three types of variable: Local, global and persistent variables.

Problem 7.

It prints `Nested`. Nested functions of the same name have precedence over subfunctions

Problem 8.

No change. `Nested` gets printed as well.