3 Loops

3.1 For loops

A for loop executes the body of the loop for every value of the index (or loop) variable. For example, the following code prints out the cubes of the numbers from 1 up to 10.

```matlab
for i=1:10
    disp(i^3);
end
```

The variable i is initially set to the first value, namely 1; the body of the loop is executed; then i is incremented to 2 and the body executed; and so on.

As another example, consider another way to calculate $\pi$: it is known that

$$1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \ldots = \frac{\pi^2}{6}$$

If we sum the first 1000 terms say, we get something that is approximately $\pi^2/6$.

```matlab
sum=0.0;
for i=1:10000
    sum = sum + 1/(i*i);
end
PIE = sqrt( 6*sum )
```

An infinite loop can be terminated by Ctrl-C.

3.2 While loops

A while loop executes the body as long as some condition remains true. Note that the condition is always checked: so the body of the loop might never be executed. Often the loop is preceded by initialization of a loop variable. For example, the following pieces of code prints out the cubes of the odd numbers from 1 up to 21.
j=1;
while j<=21
    disp(j^3);
    j=j+2;
end

3.3 Nested loops

Loops can be nested. For example, to print out multiplication table:

for i=1:10
    for j=1:10
        fprintf( '%4d', i*j );
    end
    fprintf ( '\n');
end

Sample code

Here is code to find the perfect numbers up to 10,000. A number is perfect if it equals the sum of its proper divisors.

% findPerfect
% goddard 2020
% prints perfect numbers below 10000
candidate=1;
while candidate<=10000
    sum=0;
    for i=1:candidate-1
        if mod(candidate,i)==0
            sum=sum+i;
        end
    end
    if sum==candidate
        fprintf( '%d is perfect\n', candidate);
    end
    candidate = candidate+1;
end