Θα προσπαθήσουμε να υπολογίσουμε όρους της ακολουθίας Fibonnaci:

$f_1 = 0$, $f_2 = 1$ και $f_n = f_{n-1} + f_{n-2}$, n=3,4,5,...

Θέλουμε να φτιάξουμε μια συνάρτηση που να μας επιστρέφει τον n-οστό όρο της ακολουθίας Fibonnaci (f_n).

¤Input: Integer n
Output: f_n

Method 1: File Fib1.m

```
function fn = fib1(n)
% Returns the nth number in the
% Fibonacci sequence.
f = zeros(1,n+1);
f(2) = 1;
for i = 3:n+1
    f(i) = f(i-1) + f(i-2);
end
fn = f(n);
```

Method 2: File Fib2.m

Ο πρώτος τρόπος σίγουρα σπαταλάει μνήμη αφού αποθηκεύουμε όλους τους όρους ενώ χρειαζόμαστε μόνο τον τελευταίο. Με το δεύτερο τρόπο δεν χρειάζεται να αποθηκεύουμε ολους τους όρους σε διάνυσμα.

```
function f = fib2(n)
% Returns the nth number in the
% Fibonacci sequence.
if n==1
   f = 0;
elseif n==2
   f = 1;
else
   f1 = 0;
   f2 = 1;
   for i = 2:n-1
       f = f1 + f2;
       f1=f2;
       f2 = f;
   end
end
```

epsl.m

```
clear all;
x = 1;
while 1+x > 1
    x = x/2.;
end
2*x
%fprintf(' Machine epsilon = %20.12e \n',2*x);
```

Η συνάρτηση ezplot
Μας δίνει ένα τρόπο να σχεδιάζουμε εύκολα συναρτήσεις. Το όρισμα της συνάρτησης ezplot είναι συνάρτηση.

Σύνταξη: `ezplot('fun', [a,b])` ή `ezplot(@fun, [a,b])`

```
>> ezplot('sin(3*x)',[1,3])
```

```
>> ezplot(@(x) sin(3*x),[1,3])
```

Anonymous functions

```
>> g = @(x) exp(x)-1
```

```
>> g(1)
```

Inline objects

```
>> f = inline('exp(x)-1')
```

```
>> f(1)
```

h.m

```
 function y = h(x)
%
 y = exp(x) - 1;
```

```
» h(1)
```