1)

First Order Equations:

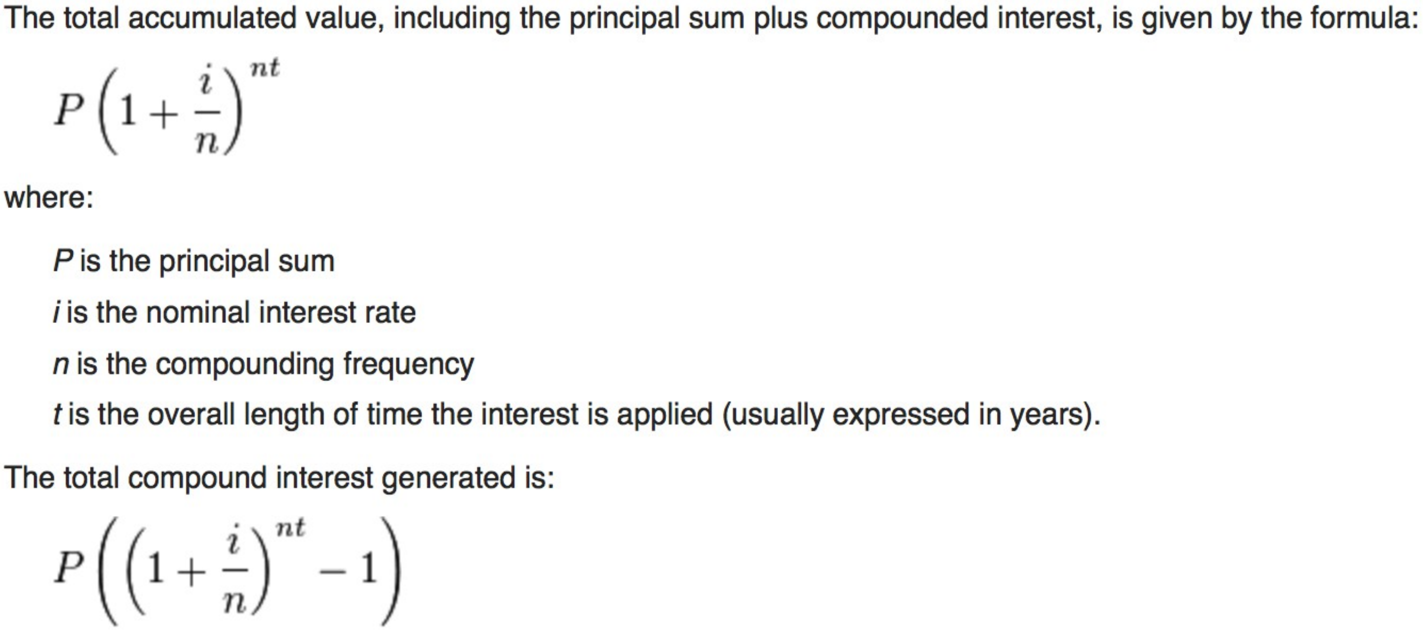
Logistic Map:

$$x_{n+1} = rx_n(1 - x_n),$$

Towers of Hanoi: A game consists of disks and stacks. You can only move one disk at a time and only move it if it is top of the stack

Hn=2Hn−1+1

Compound Interest: Compunding can be found by adding interest to the principal sum of a loan or a deposit. Compound interest is the interest of this interest.



The total accumulated value, including the principal sum plus compounded interest, is given by the formula:

$$P\left(1+\frac{i}{n}\right)^{nt}$$

where:

- *P* is the principal sum
- *i* is the nominal interest rate
- *n* is the compounding frequency
- *t* is the overall length of time the interest is applied (usually expressed in years).

The total compound interest generated is:

$$P\left(\left(1+\frac{i}{n}\right)^{nt}-1\right)$$

Second Order Equations:

Padovan Sequence: The sequence is defined by the following reccurrence relation

$$P(n) = P(n-2) + P(n-3)$$

where,

$$P(0) = P(1) = P(2) = 1.$$

Solution:

$$P(n) = \frac{1+r_1}{r_1^{n+2}(2+3r_1)} + \frac{1+r_2}{r_2^{n+2}(2+3r_2)} + \frac{1+r_3}{r_3^{n+2}(2+3r_3)},$$

where r_n is the nth root of

$$x^3 + x^2 - 1 = 0.$$

Perrin Sequence:

$$P(n) = P(n-2) + P(n-3)$$

where,

$$P(0) = 3, P(1) = 0, P(2) = 2.$$

This equation is same as the padovan sequnce but the initial conditions differ.

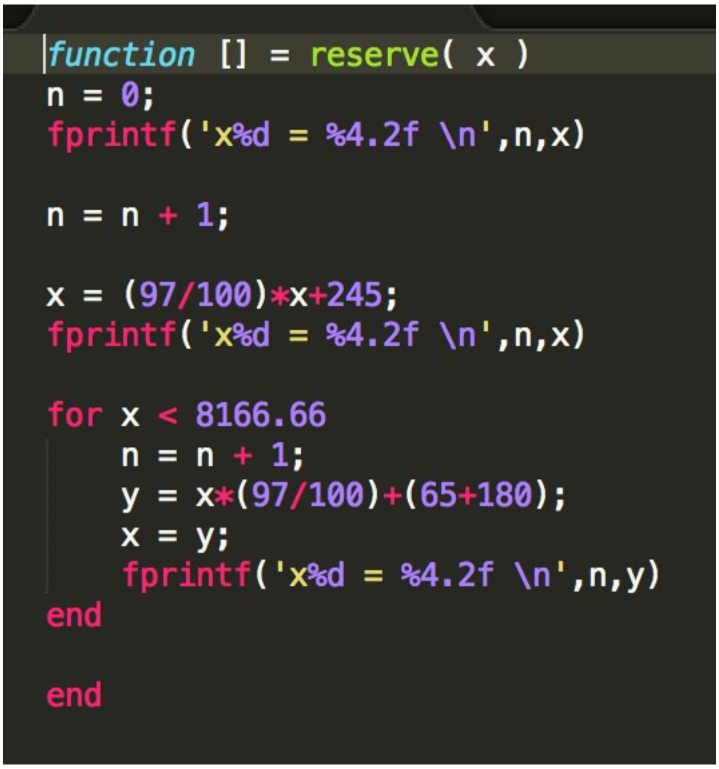
Fibonacci Number: Used for the calculatio of the golden ratio and rabbit populations.

$$F_n = F_{n-1} + F_{n-2}$$

Where F_0=0 F_1=1

Q1)
i) X_n=(97/100)*x_{n-1}+65+360
x_0 = 5000
x_1= (97/100)*x_0+65+180

ii)



```
function [] = reserve( x )
n = 0;
fprintf('x%d = %4.2f \n',n,x)

n = n + 1;

x = (97/100)*x+245;
fprintf('x%d = %4.2f \n',n,x)

for x < 8166.66
    n = n + 1;
    y = x*(97/100)+(65+180);
    x = y;
    fprintf('x%d = %4.2f \n',n,y)
end

end
```

i
ii)Stable

Q2)

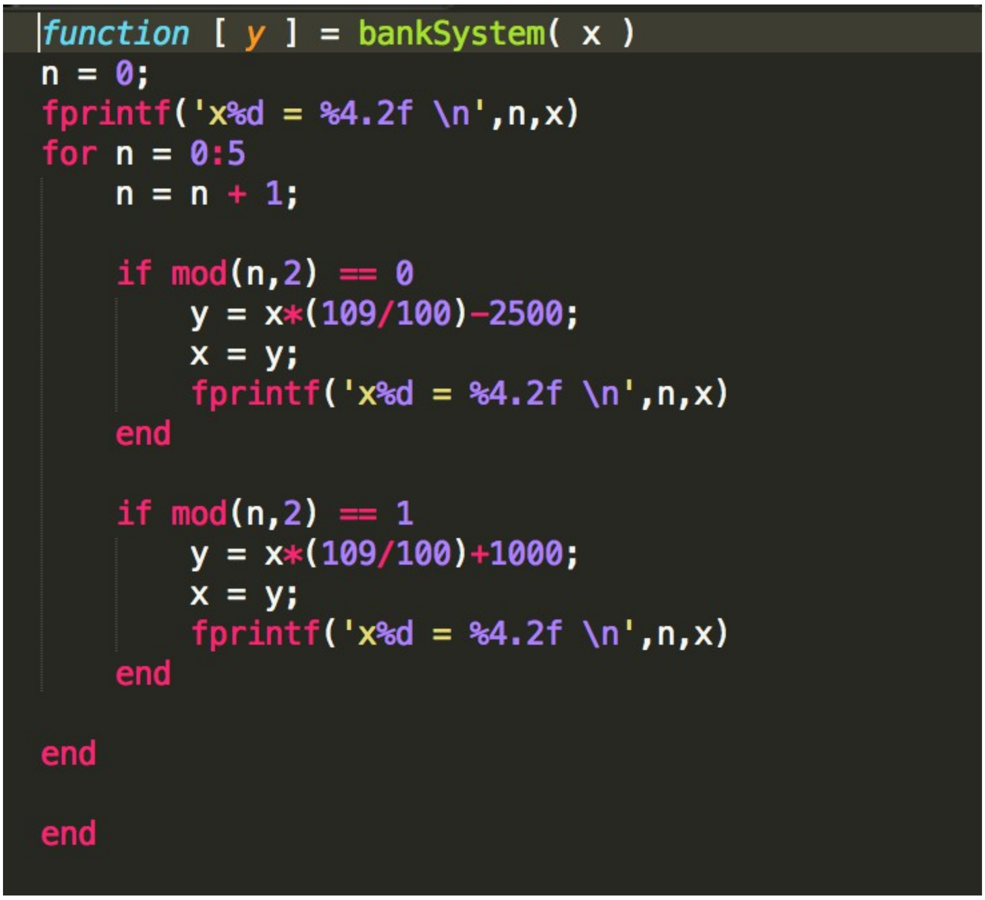
i)$X_n=x_{n-1}*(109/100)+k*1000+(k-1)*2500$

$x_0 = 40000$

k = 1 if n % 2 = 1

k = 0 if n % 2 = 0

ii)



```
function [ y ] = bankSystem( x )
n = 0;
fprintf('x%d = %4.2f \n',n,x)
for n = 0:5
    n = n + 1;

    if mod(n,2) == 0
        y = x*(109/100)-2500;
        x = y;
        fprintf('x%d = %4.2f \n',n,x)
    end

    if mod(n,2) == 1
        y = x*(109/100)+1000;
        x = y;
        fprintf('x%d = %4.2f \n',n,x)
    end

end

end
```

iii) Not Stable.