

## main.py

```
import os
from rkn import rkn

# Test Values
# m = 1
# c = 0.2
# k = 2
# a = [1,2,1.5]
# w = [0.05,1,2]

def main():

    outputText = ""
    passoInt = float(input("Passo de integração: "))
    tempoInt = float(input("Tempo total de integração: "))

    m = float(input("M: "))
    c = float(input("C: "))
    k = float(input("K: "))

    a = []
    a.append(float(input("a1: ")))
    a.append(float(input("a2: ")))
    a.append(float(input("a3: ")))

    w = []
    w.append(float(input("w1: ")))
    w.append(float(input("w2: ")))
    w.append(float(input("w3: ")))

    outputText += "Passo de Integração: " + str(passoInt) + "\n"
    outputText += "Tempo de Integração: " + str(tempoInt) + "\n"
    outputText += "m: " + str(m) + "\n"
    outputText += "c: " + str(c) + "\n"
    outputText += "k: " + str(k) + "\n"
    outputText += "a1: " + str(a[0]) + "\n"
    outputText += "a2: " + str(a[1]) + "\n"
    outputText += "a3: " + str(a[2]) + "\n"
    outputText += "w1: " + str(w[0]) + "\n"
    outputText += "w2: " + str(w[1]) + "\n"
```

```

outputText += "w3: " + str(w[2]) + "\n"

outputText += rkn(passoInt, tempoInt, m, c, k, a, w)

with open(os.path.join(os.getcwd(), "rkn.txt"), "w") as file:
    file.write(outputText)

main();

```

## rkn.py

```

import math

def getSecondOrder(params, t, y, yFirstOrder):
    [m, c, k, a, w] = params

    func = 0
    for i in range(3):
        func += a[i]*math.sin(w[i]*t)
    ySecondOrder = (func - c*yFirstOrder - k*y) / m;
    return ySecondOrder

def numberOfSteps(passoInt, tempoInt):
    return int(tempoInt/passoInt)

# ----- Factors -----
def getFactor1(t, position, velocity, h, params):
    return (h*(getSecondOrder(params, t, position, velocity) / 2))

def getFactor2(t, position, velocity, h, K, params):
    k1 = getFactor1(t, position, velocity, h, params);
    return (0.5*h*getSecondOrder(params, t + 0.5*h, position + K,
velocity + k1));

def getFactor3(t, position, velocity, h, K, params):
    k2 = getFactor2(t, position, velocity, h, K, params);
    return (0.5*h*getSecondOrder(params, t + 0.5*h, position + K,
velocity + k2))

```

```

def getFactor4(t, position, velocity, h, K, L, params):
    k3 = getFactor3(t, position, velocity, h, K, params);
    return (0.5*h*getSecondOrder(params, t + h, position + L, velocity +
2*k3))

# ----- Factors -----

# ----- Other Aux Functions -----
def getNewX(positionAtual, h, velocity, k1, k2, k3):
    return (positionAtual + h*(velocity + ((1/3)*(k1 + k2 + k3))))
#
def getBigK(velocity, h, k1):
    return (0.5*h*(velocity + 0.5*k1))
#
def getNewV(velocityAtual, k1, k2, k3, k4):
    return (velocityAtual + (1/3)*(k1 + 2*k2 + 2*k3 + k4));
# ----- Other Aux Functions -----

def createOutput(position, velocity, acceleration, t, counter):
    output = 'Iteração N°' + str(counter)
    output += '\n' + ' ' + 'Deslocamento: ' + str(position)
    output += '\n' + ' ' + 'Velocidade: ' + str(velocity)
    output += '\n' + ' ' + 'Aceleração: ' + str(acceleration)
    output += '\n' + ' ' + 'Tempo: ' + str(t)
    output += '\n\n'
    return output

def rkn(passoInt, tempoInt, m, c, k, a, w):

    t = 0
    position = 0
    velocity = 0

    params = [m, c, k, a, w]

    outputText = ''

    acceleration = getSecondOrder(params, t, position, velocity)

    n = numberOfSteps(passoInt, tempoInt)

    for counter in range(1, n):
        k1 = getFactor1(t, position, velocity, passoInt, params)

```

```
K = getBigK(velocity, passoInt, k1)
k2 = getFactor2(t, position, velocity, passoInt, K, params)
k3 = getFactor3(t, position, velocity, passoInt, K, params)
L = passoInt*(velocity + k3)
k4 = getFactor4(t, position, velocity, passoInt, K, L, params)
position = getNewX(position, passoInt, velocity, k1, k2, k3)
velocity = getNewV(velocity, k1, k2, k3, k4)
aceleration = getSecondOrder(params, t, position, velocity)
t = counter*passoInt

    outputText += createOutput(position, velocity, aceleration, t,
counter)

return outputText
```