x = [3 4 5 6 7; 1 2 4 6 8; 5 7 9 1 3; 8 7 6 5 4; 9 8 7 6 5]

total = 0

for n = 2:2:4

for k = 1:1:3

total = total + x(k,n)

end

end

total =

4

total =

6

total =

13

total =

19

total =

25

total =

26

x = 0:0.1:0.5

x =

0 0.1000 0.2000 0.3000 0.4000 0.5000

x = linspace(0,0.5,6)

x =

0 0.1000 0.2000 0.3000 0.4000 0.5000

x = [3 4 5 6 7; 1 2 4 6 8; 5 7 9 1 3; 8 7 6 5 4; 9 8 7 6 5]

x = linspace(0,0.5,6)

x =

0 0.1000 0.2000 0.3000 0.4000 0.5000

"recursion"

def k(x):

if type(x) != int or x<1:

print('x must be a positive integer')

if x==1:

return 1

else:

return x\*\*2 + k(x-1)

print(k(3))

**14**

"linear vs binary"

linear:(nonsorted)

best: 1

worst: n-1

avg: [(N+2)(N-1)]/[2n] or n/2

binary: (sorted)

best: 1

worst: log2(n)

avg: log2(n-1)

import time

start = time.clock()

stop = time.clock()

Elapsed = start – stop

diary = creates and saves a matlab file

'hold = holds the currect plot so that it can be added onto instead of covered up'

"linspace ends exactly on the value specified"

"linspace(Start,value to get to, how many values to be in array)"

"(start,step,stop)"

"M= [list(row,values or columns)]"

"divide entire row by value you want alone, minus row by previous row"

"print in matlab saves a file"

"drawnow updates a data plot"

def f(n,m):

if n >= m:

return n+m

else:

return 1 + f(n+1,m-2)

print(f(3,7))

**10**

L=[3,2,4,1]

for i in range(1,len(L)):

total = L[i] - L[i-1]

print(total)

**-1 2 -3**

L=[3,2,4,1]

for i in range(1,len(L)):

total = L[i] - L[i-1]

print(total)

**3 6 4 5**

list1 = [4,6,8]

list2 = [3,7,9,12]

for m in list2:

list1 = [m] + list1

print(list1)

**[12, 9, 7, 3, 4, 6, 8]**

x = [[0,5,2,1],[4,2,4,7],[8,9,1,2],[-2,-3,6,7]]

print(x[3][1])

print(x[1][0])

**-3 4**

for j in [3,1]:

print(x[j])

**[-2, -3, 6, 7]**

**[4, 2, 4, 7]**

for j in range(2,3):

for k in range(2):

print(x[j][k])

**8 9**

fruit = 'orange'

print(fruit[3])

print(fruit[2:5])

print(fruit[1:6:2])

**n ang rne**

def g(x):

string = '0123456789ABCDEF'

if 0<=x and x<=15:

return string[x]

else:

return

print(g(11))

**B**