import networkx as nx

import matplotlib.pyplot as plt

import random as r

def close\_dist(g,v,u,flag):

node=0

dist=999

for each in g.neighbors(u):

if flag==0 and ((u,each) in ties or (each,u) in ties):

continue

else:

d=nx.shortest\_path\_length(g,each,v)

if dist>d:

dist=d

node=each

else:

continue

return node

def myopic(g,tar,src):

path=[src]

curr=close\_dist(g,tar,src,1)

path.append(curr)

while(curr!=tar):

curr=close\_dist(g,tar,curr,0)

path.append(curr)

return len(path)-1

def homo(g,n):

ls=list(g.nodes())

for i in range(len(ls)):

g.add\_edge(ls[i],ls[i-1])

g.add\_edge(ls[i],ls[i-2])

c,d=i+1,i+2

if c==n:

c=0

if d==n:

d=0

if d==n+1:

d=1

g.add\_edge(ls[i],ls[c])

g.add\_edge(ls[i],ls[d])

def long\_tie(g):

v1=0

v2=0

ls=list(g.nodes())

while(v1==v2):

v1=r.choice(ls)

v2=r.choice(ls)

g.add\_edge(v1,v2)

ties.append((v1,v2))

def main():

g=nx.Graph()

n=input('Enter the no. of nodes:')

for each in range(n):

g.add\_node(each)

homo(g,n)

tar=input('Enter the target node:')

src=input('Enter the source node:')

x=[]

t=1

y=[]

z=[]

while(t<=n/2+1):

long\_tie(g)

x.append(t)

t+=1

y.append(myopic(g,tar,src))

z.append(nx.shortest\_path\_length(g,src,tar))

print ties

print y

print z

ties=[]

main()