```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Thu 4 April 05:12h Sweden 2019

@author: gerd doeben-henisch
Email: gerd@doeben-henisch.de
"""

###################################
# pop0d()
###################################
#
# IDEA
#
# Simple program to compute the increase/ decrease of a population with
# the parameters population number (p), birth-rate (br) and death-rate (dr).
# In this version an extension with the following features:
# - a loop to repeat the computation for n-many cycles
# - a storage of the data in an array
# - a plot of the stored data for n-many cycles
# - the ocerall change of the population in %

#########################################################
# IMPORTS
# As part of the distribution of Winpython there are already many
# libraries pre-installed, which can be activated by the import command.
# Other libraries outside of the distribution have to be downloaded
# with the pip command

import matplotlib.pyplot as plt # Lib for plotting
import numpy as np              # Lib for math

#############################################################
# DEFINITION

def pop0(p,br,dr):

    p=p+(p*br)-(p*dr)

    return p

###################################
# INPUT OF DATA


p = int(input('Population number ? '))

br = float(input('Birthrate in % ? '))
br = br/100

dr = float(input('Deathrate in % ? '))
dr = dr/100

n = int(input('How many cycles ? '))

baseYear = int(input('What is your Base Year ? '))

###############################################
```

```
# GLOBAL VARIABLES

pop = []                # storage for the pop-numbers for plotting
pop.append(p)

################################################
# COMPUTE

for i in range(n):
   p=pop0(p,br,dr)
   pop.append(p)

#####################################################
# SHOW RESULTS

for i in range(n+1):
   print('Year %5d = Citizens. %8d \n' %(baseYear+i, pop[i]) )

x = np.linspace(1,len(pop),len(pop))

plt.plot(x, pop, 'bo')

plt.show()
plt.close()

##############################################
# Compute Change of POP

n1=pop[0]
n2=pop[len(pop)-1]
Increase=(n2-n1)/(n1/100)

print(''From Year %5d, until Year  %5d, a change of %2.2f percent \n'' % (baseYear, baseYear+n,Increase) )

#####################################################
# REAL DATA
#
# UN Demographic Yearbook 2017
# https://unstats.un.org/unsd/demographic-social/products/dyb/dybsets/2017.pdf
#
# Basic Tables UN
# https://unstats.un.org/unsd/demographic-social/products/vitstats/seratab1.pdf
#
# UN public tables
# http://data.un.org/Explorer.aspx?d=POP
#
# UN Rate of population change
# http://data.un.org/Data.aspx?d=PopDiv&f=variableID%3a47
# https://www.un.org/en/development/desa/population/index.asp

'''
The br and dr values are not given explicitly in the UN data; they are inferred
from the given data and are hypothetically, but the numbers for the years 2016
and 2017 are the official numbers
The br is compiled by the fertility numbers for 2013 world wide and the overflow
by these compiled birth rates is interpreted as the dr value. This is a very weak
approach.

br=3, dr=1.9, n=15
```

Year 2016 = Citizens. 7466964

Year 2017 = Citizens. 7549100

Year 2018 = Citizens. 7632140

Year 2019 = Citizens. 7716094

Year 2020 = Citizens. 7800971

Year 2021 = Citizens. 7886781

Year 2022 = Citizens. 7973536

Year 2023 = Citizens. 8061245

Year 2024 = Citizens. 8149919

Year 2025 = Citizens. 8239568

Year 2026 = Citizens. 8330203

Year 2027 = Citizens. 8421835

Year 2028 = Citizens. 8514475

Year 2029 = Citizens. 8608135

Year 2030 = Citizens. 8702824

Year 2031 = Citizens. 8798555

$^{O}{}_{B_J}$

From Year 2016, until Year 2031, a change of 17.83 percent

$^{O}{}_{B_J}$

From Year 2016, until Year 2031, a change of 17.83 percent

'''