1. The full Python inputs and results are shown below. Please note that there may be some slight rounding errors present, and that the results may not be formatted exactly as you would find in Python.

Here is the program:

```
from __future__ import division
from pylab import *

# Defining the ydash function
def ydash(y):
    ans = 3.86*y
    return ans

# The minimum and maximum t values
tMin = 0
tStep = 1
tMax = 17

# Increasing the value of tMax so that the array size is correct
tMax = tMax+1

# The t values
ts = arange(tMin, tMax, tStep)
# The number of t points
numSteps = size(ts)

# Creating the array to store the y values
ys = zeros(numSteps, dtype=float)
ys[0] = 5

# Indexing variable to traverse the arrays
i = 1

# Adding headers for the table of results
print "t", "y"

# Looping through all the t values
while i < numSteps :
    # Applying Euler's method
    ys[i] = ys[i-1]+tStep*ydash(ys[i-1])

    # Creating a table of values
    print ts[i], ys[i]

    # Incrementing the indexing variable
    i = i+1

# Print a blank line to separate the values
print
print "The final population size is:", ys[numSteps-1]
```

Here is the output from running the program:

```
t y
```

```
1 24.3
2 118.098
3 573.95628
4 2789.42752
5 13556.6178
6 65885.1623
7 320201.889
8 1556181.18
9 7563040.53
10 3.675638e+07
11 1.786360e+08
12 8.681709e+08
13 4.219311e+09
14 2.050585e+10
15 9.965843e+10
16 4.843400e+11
17 2.353892e+12

The final population size is: 2.353892e+12
```

2. The full Python inputs and results are shown below. Please note that there may be some slight rounding errors present, and that the results may not be formatted exactly as you would find in Python.

Here is the program:

```
from __future__ import division
from pylab import *

# Defining the ydash function
def ydash(y):
    ans = 3.89*y
    return ans

# The minimum and maximum t values
tMin = 0
tStep = 1
tMax = 10

# Increasing the value of tMax so that the array size is correct
tMax = tMax+1

# The t values
ts = arange(tMin, tMax, tStep)
# The number of t points
numSteps = size(ts)

# Creating the array to store the y values
ys = zeros(numSteps, dtype=float)
ys[0] = 38

# Indexing variable to traverse the arrays
i = 1

# Adding headers for the table of results
print "t", "y"
```

```
# Looping through all the t values
while i < numSteps :
    # Applying Euler's method
    ys[i] = ys[i-1]+tStep*ydash(ys[i-1])

    # Creating a table of values
    print ts[i], ys[i]

    # Incrementing the indexing variable
    i = i+1

# Print a blank line to separate the values
print
print "The final population size is:", ys[numSteps-1]
```

Here is the output from running the program:

```
t y
1 185.82
2 908.659800
3 4443.34642
4 21727.9640
5 106249.744
6 519561.248
7 2540654.50
8 1.242380e+07
9 6.075238e+07
10 2.970792e+08

The final population size is: 2.970792e+08
```

3. The full Python inputs and results are shown below. Please note that there may be some slight rounding errors present, and that the results may not be formatted exactly as you would find in Python.

Here is the program:

```
from __future__ import division
from pylab import *

# Defining the ydash function
def ydash(y):
    ans = 2.21*y
    return ans

# The minimum and maximum t values
tMin = 0
tStep = 1
tMax = 14

# Increasing the value of tMax so that the array size is correct
tMax = tMax+1

# The t values
ts = arange(tMin, tMax, tStep)
# The number of t points
```

```
numSteps = size(ts)

# Creating the array to store the y values
ys = zeros(numSteps, dtype=float)
ys[0] = 41

# Indexing variable to traverse the arrays
i = 1

# Adding headers for the table of results
print "t", "y"

# Looping through all the t values
while i < numSteps :
    # Applying Euler's method
    ys[i] = ys[i-1]+tStep*ydash(ys[i-1])

    # Creating a table of values
    print ts[i], ys[i]

    # Incrementing the indexing variable
    i = i+1

# Print a blank line to separate the values
print
print "The final population size is:", ys[numSteps-1]
```

Here is the output from running the program:

```
t y
1 131.61
2 422.468100
3 1356.12260
4 4353.15355
5 13973.6229
6 44855.3295
7 143985.608
8 462193.801
9 1483642.10
10 4762491.14
11 1.528760e+07
12 4.907318e+07
13 1.575249e+08
14 5.056550e+08

The final population size is: 5.056550e+08
```