function: change_prey

1. Inputs: birth rate, predation rate, initial prey population, initial predator population
2. Output: change in the prey population
3. This is a simple mathematical calculation. It is calculated by subtracting the prey that are eaten from the number of new prey born.
    a. birth * prey - predation * prey * pred

function: change_pred

1. Inputs: growth rate, death rate, initial prey population, initial predator population
2. Output: change in the predator population
3. This is a simple mathematical calculation. It is calculated by subtracting the predators that die from the growth in the predator population.
    a. growth * prey * pred - death * pred

function: pred_prey_intervals

1. Inputs: birth rate, predation rate, growth rate, death rate, initial populations, number of years, size of interval
2. Output: elapsed time and new predator/prey populations
3. The starting elapsed time is set to 0.0 and a list is created that will hold the data of outputs in groups of 3 (time, prey, pred).
4. A list is created to hold each triplet through future appendation.
5. The number of intervals is calculated from intervals * years.
6. Every value in this range represents one interval of time. For every value in this range, the elapsed time is calculated by taking the first element of triplets. For every value in this range, the new prey population is calculated by using the change_prey function with the inputs. For every value in this range, the new pred population is calculated by using the change_pred function with the inputs.
7. The float function is used to keep data precise.
8. A new triplet containing the new time and populations is appended to the created list after each iteration.
9. The full list is returned after appendation is finished.

function: plot_pred_prey

1. Inputs: population data, names of prey/predators
2. The simpleplot module is imported to allow plotting functions to be accessible.
3. An empty list is created for future appendation.
4. For every set of 3 in the input population data (in the order time, prey, pred), only the prey and pred parts are grouped together and appended to the empty list.
5. Output: The predator data (y-axis) is plotted against the prey data (x-axis).

function: plot_time_populations

1. Inputs: population data, names of prey/predators
2. Two empty lists are created for future appendation. One for predator data, one for prey.
3. For each set of triples (time, prey, pred), the time and prey are grouped together and appended to the prey list.
4. For each set of triples (time, prey, pred), the time and pred are grouped together and appended to the pred list.

5. Output: Both sets of data are plotted in respect time.

function: pred_prey_carrying

1. Inputs: birth rate, predation rate, growth rate, death rate, initial populations, number of years, size of interval, carrying capacity
2. Output: elapsed time and new capped predator/prey populations
3. The starting elapsed time is set to 0.0 and a list is created that will hold the data of outputs in groups of 3 (time, prey, pred).
4. A list is created to hold each triplet through future appendation.
5. The number of intervals is calculated from intervals * years.
6. Every value in this range represents one interval of time. For every value in this range, the elapsed time is calculated by taking the first element of triplets. For every value in this range, the new prey population is calculated by using the change_prey function with the inputs. For every value in this range, the new pred population is calculated by using the change_pred function with the inputs.
7. However, if the prey population every exceeds the set carrying capacity, the value of the prey population is reset to the carrying capacity.
8. The float function is used to keep data precise.
9. A new triplet containing the new time and populations is appended to the created list after each iteration.
10. The full list is returned after appendation is finished.

1. The prey population continually increases.
   a. (10, 0, 0, 0, 100, 0, 50, 100)
   b. The predation rate must exceed the birth rate.
   c. Growth rate, death rate, and initial predator population do not matter.
   d. Initial prey population must be a positive number that allows for reproduction.
   e. The years and intervals are chosen in accordance with assignment recommendations.
   f. In this example, there are no predators, so initial predator population, predation rate, growth rate, and death rate are all 0.
2. The predator population continually increases.
   a. (5, 5, 10, 5, 100, 100, 50, 100)
   b. The growth rate must exceed the death rate.
   c. Birth rate, predation rate, and initial prey population do not matter.
   d. Initial predator population must be a positive number that allows for growth.
   e. The years and intervals are chosen in accordance with assignment recommendations.
3. The prey population continually decreases.
   a. (5, 10, 5, 5, 100, 100, 50, 100)
   b. The predation rate must exceed the birth rate.
   c. Growth rate, death rate, and initial predator population do not matter.
   d. Initial prey population must be a positive number that allows for reproduction.
   e. The years and intervals are chosen in accordance with assignment recommendations.
4. The predator population continually decreases.

a. (5, 5, 5, 10, 100, 100, 50, 100)
b. The death rate must exceed the growth rate.
c. Birth rate, predation rate, and initial prey population do not matter.
d. Initial predator population must be a positive number that allows for growth.
e. The years and intervals are chosen in accordance with assignment recommendations.

Adding a carrying capacity will cause the graph to reach a plateau after its period of exponential growth. While without a carrying capacity large growth may have been continually observed, the carrying capacity ensures that the shape of the curve will reduce in slope and plateau off.