Code	**GENETIC_ALGORITHM.PY**
Author	Nathaniel Heatwole, PhD (heatwolen@gmail.com) (GitHub) (LinkedIn)
Summary	Use a genetic algorithm to optimize (maximize) an erratic function, one that has many sharp peaks/valleys and local extrema and several "distractor" answers

Methods/ Process	Genetic algorithm
	- Iterative optimization method that mimics the workings of biological evolution
	- As the iterations progress, the pool of candidate solutions generally improves
	- Can optimize complex parameter spaces (including non-convex, non-differentiable, and non-continuous)
	- Concepts:
	  - *Fitness*: objective function ("genetic" representation of parameter space)
	  - *Population*: total simulated people
	  - *Selection*: randomly select persons from current population to breed (with replacement), in proportion to their fitness
	  - *Crossover*: randomly combine parameter values ("genes") from parents
	  - *Mutation*: randomly replace some genes with randomly chosen values (helps ensure good areas of the solution space are not overlooked)
	  - *Generations*: total iterations
	  - *Output*: global best solution (across all generations)

Objective Function	Maximize:[1] Y(X) = | sin(A * X) + sin(B * X) + sin(C * X) | with: A = (2*π) / 13 B = (2*π) / 18 C = (2*π) / 23
Results	Algorithm efficiently locates the neighborhood of the maximum value

[1] This function is similar to the one at https://mathblag.wordpress.com/2013/09/01/sums-of-periodic-functions.