German University in Cairo GUC
Faculty of Media Engineering and Technology
Prof. Dr. Mohammed Abdel-Megeed Salem
Eng. Mariam Mohamed Fawzy

Computational Intelligence and Neural Networks(CSEN1121)
Winter Semester 2025

In this assignment, you are required to implement a genetic algorithm for Flower Evolution from scratch. The objective is to simulate the evolutionary process of a population of flowers across multiple generations. The flowers will evolve based on fitness, which is determined by user interaction. This project will help you understand the basic principles of genetic algorithms, including selection, crossover, and mutation.

The Following Outlines the Essential Requirements that your Implementation Must Fulfill.

1) **Create a Population of Flowers**: Develop a population of "flowers," where each flower has a DNA sequence of 11 genes that determines its characteristics:
    i. **Size of the center**: Represented by a single gene
    ii. **Color of the center**: Represented by three genes (Red, Green, Blue), each with values ranging from 0 to 255. These values represent the intensity of the center's color
    iii. **Color of the petals**: Represented by three genes (Red, Green, Blue), each with values ranging from 0 to 255. These values represent the intensity of Petal's color
    iv. **Color of the stem**: Represented by three genes (Red, Green, Blue), each with values ranging from 0 to 255. These values represent the intensity of Stem's color
    v. **Number of petals**: represented by a single gene with values ranging from 0 to 7

You may create a class to represent individual flowers, which will be displayed based on the encoded genetic information (DNA).

2) **Initialize the Population**: Write a function to initialize a population of flowers with random DNA sequences. The population size is 8 flowers.

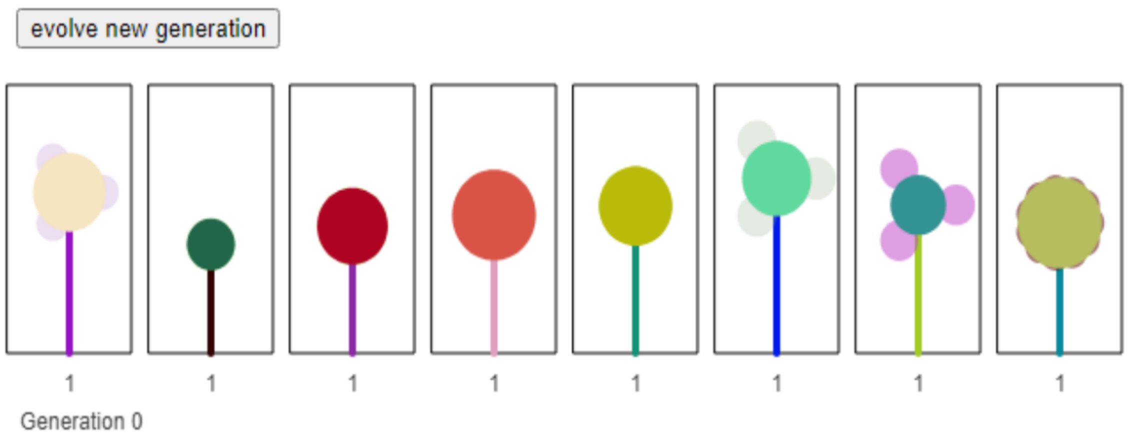
3) **Implement Core Genetic Algorithm Functions**:
    - **Selection**: Choose individuals (flowers) for reproduction based on their fitness. The fitter a flower, the more likely it is to be selected
    - **Crossover**: Combine the genetic information of two parent flowers to create a new child flower. This simulates reproduction. The crossover occurs with a 65% probability
    - **Mutation**: Randomly change parts of a flower's DNA to introduce variability into the population. In this simulation, mutation occurs with a 5% probability (mutation rate is 0.05)

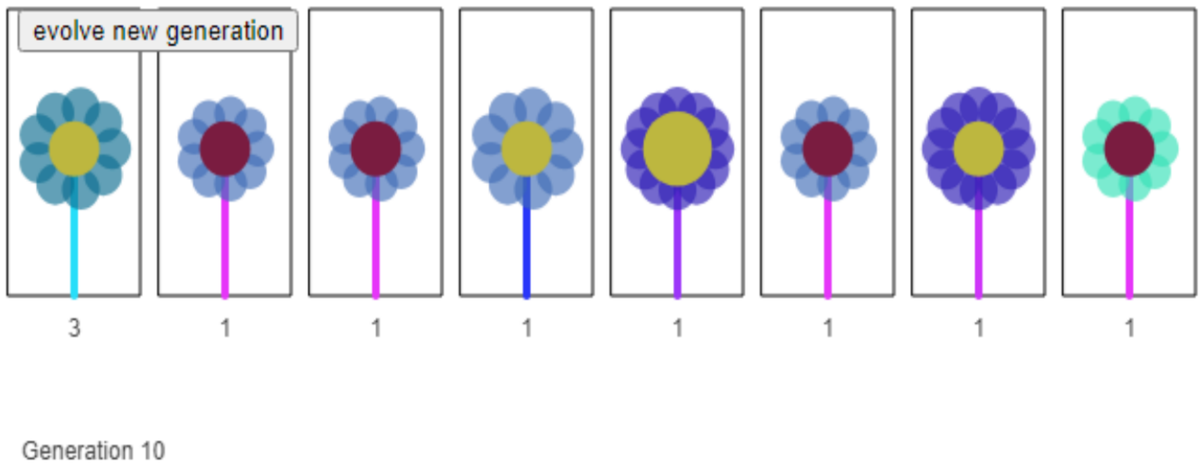
4) **User Interaction to Measure Fitness:** The fitness of each flower will be determined based on how long the user hovers over it with the mouse. Flowers that are hovered over for longer periods will be considered fitter.

5) **Evolve the Population:** At each generation, select parents, create offspring through crossover, apply mutation, and replace the old population with the new one. Over successive generations, the population should evolve to have flowers that resemble fitter individuals.

Demonstration of the Expected Output

The implementation should demonstrate the successful evolution of a flower population over successive generations, where the evolutionary process is influenced by user interaction. Specifically, the program must clearly show how flower characteristics change and adapt as generations progress, reflecting the underlying genetic algorithm principles. You should see fitter flowers (those hovered over more) influencing the next generation.





To ensure clarity and traceability of the evolutionary process, the program output must also include detailed print statements that display intermediate steps such as:

- **Population**: A list of the generated flowers with their characteristics and associated fitness
- **Population after Sorting**: The same population ordered by fitness values
- **Selected Flowers for Reproduction**: The subset of flowers chosen as parents
- **Crossover Results**: For each reproduction step, the program should print the two parent flowers (before crossover) and the resulting child flowers (after crossover).
- **Mutation Results**: The characteristics of the flowers before and after applying the mutation
- **Updated Population**: The new generation of flowers after selection, crossover, and mutation are applied.

These print statements must be included for every generation so that changes in flower characteristics can be traced and verified over time.

Regulations and Deliverables:

1) Your Code should handle all Exceptional Cases
2) The deadline of submitting the assignment is **Thursday, 23 October 2025, at 11:59 PM**
3) Submit your work to csen1121.winter2025@gmail.com
4) The Subject to the Email is CINN_Assign1_[ID1_ID2_ID3]
5) A deduction of 5% for the late submissions till 5:59 am the next morning.
6) The deliverables are (in one zip file named as CINN_Assign1_[ID1_ID2_ID3]):
    1. **Code**: `.ipynb file` named CINN_Assign1_[ID1_ID2_ID3]
    2. **Report**: `.pdf file` named CINN_Assign1_[ID1_ID2_ID3] explaining the implementation details and explanation, team members' details and contributions, and snapshots of the results (simulation and the print statements)
7) Plagiarized or AI-generated code is unacceptable and will result in a ZERO grade
8) Evaluation will be held for this assignment. Your grade is influenced by your performance in the evaluation
9) By reserving an evaluation slot for the assignment, you are committing to attend at the scheduled time. If you fail to show up on time or do not attend at all, you will receive a grade of ZERO for the evaluation. No rescheduling will be allowed after a missed appointment.