// COMSC210 | Lab 29 | Winston Jose
// Project Proposal
// Github Link: https://github.com/winstonjose01/COMSC210-Lab29-bugslife

Project Title: It's A Bug's Life

Project Proposal
This project aims to simulate population dynamics in a small ecosystem with aphids, ants, and ladybugs. Each insect population fluctuates based on environmental factors such as temperature, precipitation, and UV index, as well as interactions between the species. For instance, ladybugs prey on aphids, ants protect aphids in exchange for a sugary secretion, and these relationships affect population sustainability. By modeling these interactions, the simulation provides insight into how environmental conditions and species interactions influence ecosystem stability.

Implementation Method
The simulation uses a `map` to manage insect populations, with each species (aphids, ants, ladybugs) as keys and **arrays** as values. Each array has a daily list of populations for specific time periods, updated based on environmental factors. A `list` stores historical population data, allowing tracking over time and enabling comparisons. For example, `map<string, array<list<int>, 3>>` might represent the populations for each species, with each array storing lists of daily population counts, one for each environmental influence.

Simulation Events and Modeling
1. Insect Population Affected by Temperature: Temperature fluctuations increase or decrease aphid reproduction and, indirectly, the populations of ants and ladybugs through their food supply.
2. Insect Population Affected by Precipitation: Higher precipitation can reduce UV impact, benefiting aphids by promoting plant growth, but may disrupt ant activity. Water can penetrate the anthill colony disrupting their daily activity and foraging for food. Ladybugs avoid flying in the rain and also hampers their ability to look for aphids
3. Insect Population Affected by UV Index: Higher UV exposure decreases aphid survival, indirectly affecting ants (which rely on aphids) and ladybugs (which consume aphids).
4. Insect Symbiotic Relations. Ladybugs prey on aphids, reducing their population, while ants protect aphids for their honeydew secretion, creating mutual dependencies that balance or strain the populations depending on environmental conditions.

This project attempts to create a simplified model of interspecies interactions between aphids, ants and ladybugs under changing environmental variables.

Pseudo-code:

// —----------Aphid.h

Include any headers for math and string

Class Aphid{

private:
int P_aphid_t0 = variable for initial **aphid** population;
int P_aphid_t1 = variable for current **aphid** population;
float aphid_GR = an **aphid's** typical population growth rate
floatphid_MR_Temp = an **aphid's** mortality rate when affected by temperature
float aphid_MR_UV = an **aphid's** mortality rate when affected by **UV index**
float aphid_Topt = the optimum **temperature** for health **aphid** growth
float aphid_Pred_LB = factor that defines the **predatory** effect between **aphids and ladybugs**
float K_benefit = factor that defines the symbiotic effect between aphids and ants
public:
Aphid constructor (int) {
// Set the initial population of aphids }
set_initial_pop() function {
//sets the initial population of aphids (Population the day before) }
get_initial_pop() function {
//gets and returns the he initial population of the of aphids (Population the day before) }
calc_current_pop() {
// function calculates the aphid population based on input factors
P_aphid_t1 = current aphid population for this time interval }
get_current_pop() { function gets and returns the new calculated population }
}

//—---------- Ant.h ----------

Include any headers for math and string

Class Aphid{

private:

int P_ant_t0 = variable for initial **ant** population;
int P_ant_t1 = t variable for current **ant** population;
float ant_GR = factor that defines **ant** typical growth rate
float ant_MR_Temp = factor that defines **ants** mortality rate when affected by temperature
float ant_MR_Precip = factor that defines **ants** mortality rate when affected by **precipitation**
float ant_Topt = factor that defines the optimum **temperature** for healthy **ant**
float K_benefit = factor that defines the **symbiotic** effect between **aphids and ants**

public:

Ant constructor {
Set the initial population of **ants**)}
set_initial_pop() function {
//sets the initial population of ants (Population the day before) }
get_initial_pop() function {
//gets and returns the initial population of the of ants (Population the day before) }
calc_current_pop() {
// function calculates the ant population based on input factors
P_ant_t1 = current aphid population for this time interval }
get_current_pop() { function gets and returns the new calculated population }

}

//—---------- Ladybug.h

Include any headers for math and string

Class Aphid{

private:

int P_ladybug_t0 = variable for initial **ladybug** population;
int P_ladybug_t1 = variable for current **ladybug** population;
float ladybug_GR = factor that defines **ladybug** typical growth rate
float ladybug_MR_Temp = factor that defines **ladybug** mortality rate when affected by **temperature**
float ladybug_MR_UV = factor that defines **ladybug** mortality rate when affected by **UV**
float ladybug_Topt = factor that defines the optimum **temperature** for healthy **ant**
float ladybug_Pred_aphid = factor that defines the **predatory** effect between **aphids and ladybugs**

public:

Ladybug constructor {
Set the initial population of **ants**)}
set_initial_pop() function {
//sets the initial population of ladybugs (Population the day before) }

```
        get_initial_pop() function {
                //gets and returns the initial population of the of ladybugs (Population the day
        before) }
        calc_current_pop() {
                // function calculates the ant population based on input factors
                P_ladybug_t1 = current aphid population for this time interval }
        get_current_pop() { function gets and returns the new calculated population }

}

//—------------------------------------------------------------ main.cpp
-------------------------------------------------

# Include any headers for file handling, data structures, etc.
#include <iostream>, <fstream>,<sstream>,<string> , <array>, <map>, <list>

// Constant variables to

        const int DAYS_IN_MONTH[] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
        const string MONTH[] = {"Jan","Feb","march",
                                "Apr","May","Jun",
                                "Jul","Aug","Sep",
                                "Oct","Nov","Dec"};

// Function Prototype
        map<string,array<float,3>> load_environment_factors // Function to read and load
factors

// Main Function
int main(){

        int aphid_pop;          // Aphid population
        int ant_pop;            // Ant population
        int ladybug_pop         // Ladybug population


        Aphid aphid(1000);    // Create Aphid object and initialize with a starting population
        Ant ant(400)            // Create Ant object and initialize with a starting population
        Ladybug ladybug(300) // Create Ant object and initialize with a starting population

        map<string, array<list<int>,3>> population_result;  // Declare the data structure to use
for simulation

for each month in MONTH[]:
```

```
{
        for each day in DAYS_IN_MONTH[]:
        {
                // Read the file and load the enviromental factors
                 temperature, precipitation, uv_index = environment_factors[date]

                aphid_pop =  aphid.get_initial population // Get the initial population of the aphids
                 ant_pop = ant.get_initial population // Get the initial population of the ants
                ladybug_pop = ladybug.get_initial population // Get the initial population of the
        ladybugs

                // Calculate the population of each insect for the current date iteration and pass
        temperature, UV index, precipitation, ladybug and ant population
                aphid.set_population({temperature, precipitation, uv_index, ladybug_pop,
        ant_pop})

                // Calculate ant population and pass temperature, precipitation and aphid
        population
                ant.calculate_population({temperature, precipitation, uv_index})

                // Calculate ladybug current population and pass temperature, UV index and
        aphid population
                ladybug.calculate_population({temperature, uv_index, aphid_pop})

                // Update the current population of each insect
                 aphid.set_current_population(new_population)
                 ant.set_current_population(new_population)
                ladybug.set_current_population(new_population)

                // Store the calculated population i the map
                populationResults[date][0].push_back(aphid.getPopulation());
               populationResults[date][1].push_back(ant.getPopulation());
              populationResults[date][2].push_back(ladybug.getPopulation());
        }

        // Print population data for verification
        for ( [date,populations] : populationResults) {
                    std::cout << date << " - Aphid Population: " << populations[0]  // Print aphid
        population
                              << ", Ant Population: " << populations[1]                    // Print ant
        population
                              << ", Ladybug Population: " << populations[2] << std::endl; //Print
        ladybug population
                }
```

```
}

// function to load the environmental factors from the text file
// arguments: a string filename
// returns a map with an array of 3 float numbers

map <string, array<float,3>>> load_environment_factors function (string filename){

        map<string, array<float, 3>> environment_factors // map to load the temperature
precipitation, UV index

        ifstream file (filename)  // Create a filestream
        string line, month         // Create variables for reading the line and month
        int day                       // Create variable to store the day
        float temperature, precipitation, uv_index // Create variable to store the environment
values

        while (geline(file, line)) // Read the lines in the file {
                isstring stream (line)   // treat the line as an input stream
                getline (stream, month,",")
                stream >> day >> temperature >> precipitation >> UV index

                //concatenate month and day as a string to be the key
                date = month + string(day)

                // Store values into the map environment factors
                environment_factors[date] = {temperature, precipitation, uv_index};
        }
        return environment_factors;
```