Programming Exercises

The following require you to open the skeleton program and make modifications.

Question 1

This question refers to `SimulateSpring`.

Currently, the source code runs in such a way that a seed *always* turns into a plant during summer. Modify `SimulateSpring` so that each seed has only a 40% chance of becoming a plant. Seeds that do not become plants remain in the ground and may become plants in later years.

Evidence you need to provide: **(4 marks)**

- Your amended SOURCE CODE PROGRAM for `SimulateSpring`
- One SCREEN CAPTURE, from the spring of year 1, having loaded the data file during execution

Question 2

This question refers to `ReadFile`.

At present, the user must include the suffix .txt in order to load a file. Modify the program so that '.txt' is automatically added, meaning the user is not required to include the '.txt' suffix. If, however, the user *does* include the .txt suffix, nothing will be automatically added to the user's input.

Evidence you need to provide: **(4 marks)**

- Your amended SOURCE CODE PROGRAM for `ReadFile`
- One SCREEN CAPTURE, showing entry of the data's correct file name with the '.txt' suffix, along with the output for spring of a one-year simulation
- One SCREEN CAPTURE, showing entry of the data's correct file name without the '.txt' suffix, along with the output for spring of a one-year simulation

Question 3

This question refers to `Simulation`.

The first field the program outputs is always spring of the first year. Modify the program so that, before the 'spring' layout is output, the starting state of the field is output, i.e. the state of the field before spring of the first year.

There should be a call to `Display` from `Simulation`, which will pass, as parameters, the integer '0' in place of a year, and the string 'start' in place of a season. You should make no changes to `Display`.

Evidence you need to provide: **(3 marks)**

- Your amended SOURCE CODE PROGRAM for `Simulation`
- One SCREEN CAPTURE, showing the full output of the program for which '0' years were entered at the menu, and the file was loaded

Question 4

This question refers to `SimulateSummer`.

Presently, when there is a severe drought, the program simply outputs "There has been a severe drought". Modify the code so that it also outputs the number of plants that have died, in the following format:

"There has been a severe drought: X plants have died".

Evidence you need to provide: **(4 marks)**

- Your amended sections of the SOURCE CODE PROGRAM for `SimulateSummer`

Question 5

This question refers to `CreateNewField`.

Currently, a new field will consist of only a single seed, in the middle of the field, and no rocks. Modify the program so that, for a new field, five rocks are generated and placed at random. If an attempt is made to place a rock where the seed, or another rock, already exists, a new random position should be generated for that rock.

Evidence you need to provide: **(6 marks)**

- Your amended SOURCE CODE PROGRAM for `CreateNewField`
- One SCREEN CAPTURE, showing a user input requesting one year, along with the output for spring of the first year

Question 6

This question refers to `GetHowLongToRun`.

The user is currently prompted for a number between -1 and 5, although the program allows the entry of any other integers, and crashes when the user attempts to enter any nun-numerical input. Add data validation code that will cause the program to loop until an integer between -1 and 5 has been entered on the first menu.
An error message of "Please enter a whole number between -1 and 5" should be output if the input is invalid.

Evidence you need to provide: **(9 marks)**

- Your amended SOURCE CODE PROGRAM for `GetHowLongToRun`
- One SCREEN CAPTURE, showing an attempt to add the character 'x' in response to the prompt "Enter a number between 0 and 5, or -1 for stepping mode:".
- One SCREEN CAPTURE, showing an attempt to add the integer 6 in response to the prompt "Enter a number between 0 and 5, or -1 for stepping mode:".
- One SCREEN CAPTURE, showing an attempt to add the integer '1' in response to the prompt "Enter a number between 0 and 5, or -1 for stepping mode:".

Question 7

This question refers to `SimulateSpring` and to the constants section.

A new type of plant is to be introduced to the simulation, which will be a weed, requiring a constant, called `WEED`, which stores a capital 'W'. Every spring, each point containing soil has a 1 in 10 chance of growing a weed. Weeds, unlike plants, do not die in winter.

Evidence you need to provide: **(6 marks)**

- Your amended SOURCE CODE PROGRAM for the constants section
- Your amended SOURCE CODE PROGRAM for `SimulateSpring`
- One SCREEN CAPTURE, showing the spring output in year 1 of a two-year simulation (using a new field, not the data file)
- One SCREEN CAPTURE, showing the spring output in year 2 of a two-year simulation (using a new field, not the data file)

Question 8

This question refers to `CreateNewField`.

When a new field is created, a single seed is placed at the centre of that field. Modify the code so that the user is prompted for a number of seeds that will be randomly dispersed throughout the field. The code that plants the seed in the middle of the field should be removed.

If a seed is randomly placed in a position where a seed already exists, the second seed dies. Before the output for spring of the first year, the program should output the number of seeds that have died in this way in the following format:

"x seed(s) lost."

Evidence you need to provide: **(11 marks)**

- Your amended SOURCE CODE PROGRAM for `CreateNewField`
- One SCREEN CAPTURE, showing a user input requesting 20 seeds, along with the output for spring of the first year (new field, one-year simulation)

Question 9

This question refers to `InitialiseField`.

As the program currently runs, it requires the user to enter a capital 'Y' in order to load a file. If anything else is entered, a new field is created. The following changes should be made to how the program runs:

- Entering either an upper case 'Y' or a lower case 'y' should result in the user being prompted for a filename, by `ReadFile` being called
- Entering an upper case 'N' or a lower case 'n' should result in a new field, by `CreateNewField` being called
- Entering anything else should result in an appropriate error message, with the user then prompted with the original question again

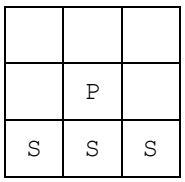
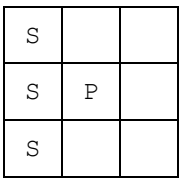
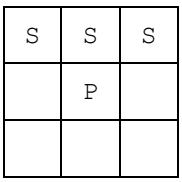
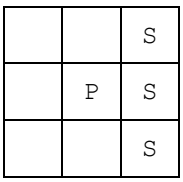
Evidence you need to provide: **(8 marks)**

- Your amended SOURCE CODE PROGRAM for `InitialiseField`
- One SCREEN CAPTURE, showing the result of entering a lower case 'y' in response to the question "Do you want to load a file with seed positions?"
- One SCREEN CAPTURE, showing the result of entering a lower case 'n' in response to the question "Do you want to load a file with seed positions?"
- One SCREEN CAPTURE, showing the result of entering a lower case 'x' in response to the question "Do you want to load a file with seed positions?"

Question 10

This question refers to `SimulateAutumn`.

In autumn, wind will blow in from the north, south, east or west. During `SimulateAutumn`, a random integer should be generated that will be 0, 1, 2 or 3. That number will cause seeds to be generated in a particular way, indicated as follows:

Random number	0	1	2	3
Wind direction	From North	From East	From South	From West
Seed dispersal				

Evidence you need to provide: **(9 marks)**

- Your amended SOURCE CODE PROGRAM for `SimulateAutumn`
- One SCREEN CAPTURE, from the autumn of year 1, run from a new field, showing all output relating to autumn

Question 11

This question refers to `Simulation`, as well as a new procedure, `SaveField`.

When a simulation has ended, after the output "End of Simulation", the user should be prompted as to whether they wish to save the field. If they do, the `Simulation` procedure should call a new procedure named `SaveField`, which will require a two-dimensional character array, `Field`, to be passed to it as a parameter. `SaveField` will perform three tasks:

- Prompt the user for a file name, to which the program will add a '.txt' suffix
- Attempt to save the field in this file, with each field row on a separate row in the text file
- If the save is successful, display the message "File saved", otherwise, display "File error".

Evidence you need to provide: **(18 marks)**

- Your amended SOURCE CODE PROGRAM for `Simulation`
- The SOURCE CODE for a new procedure, in full, called `SaveField`
- One SCREEN CAPTURE that shows the following, after the simulation has run for one year:
  - The user indicating that they would like to save the field
  - The user entering 'test' as the file name
  - The program's subsequent output
- One SCREEN CAPTURE that shows the output for spring, having loaded 'test.txt' and run the simulation for one year

Question 12

This question refers to `SimulateSummer`, a new function, `GrowPlants`, and the constants section.

Currently, if a drought occurs, some of the plants within the simulation are killed, but nothing happens during summer in the event that there is no drought. Modify the program to enable the following:

- If there is a drought, the program should function as it currently functions
- If there is no drought, the program should call a new function, `GrowPlants`, once for each plant in the field
- `GrowPlants` requires the following parameters:
  - A two-dimentional character array, `Field`
  - An integer, `Row`, indicating the row the of the current plant
  - An integer, `Column`, indicating the column of the current plant
- `GrowPlants` should return a two-dimensional character array
- For each directly adjacent point to the plant on the field (i.e. each of north, south, east and west), there should be a 25% chance of an offshoot, which will be represented as a capital 'O' on the field and should use a constant, `OFFSHOOT`
- Offshoots can only sprout in locations currently containing only soil
- Offshoots cannot sprout outside the field for a plant on the edge of the field
- At the start of `SimulateSummer`, any offshoots from last year become plants, before the possibility of drought is considered

Evidence you need to provide: **(16 marks)**

- Your amended SOURCE CODE PROGRAM for `SimulateSummer`
- Your amended SOURCE CODE PROGRAM for the constants section
- The SOURCE CODE for a new procedure, in full, called `GrowPlants`

Question 13

This question refers to multiple sections of the skeleton code.

Currently, any seeds in the ground at the beginning of spring become plants during spring. You are going to modify the program to model a seed that takes longer to germinate. The constant `SEED` will no longer exist, and will be replaced by two new constants, `NEWSEED` (storing a lower case 's') and `MATURESEED` (storing an upper case 'S').

During spring, new seeds will become mature seeds, and will remain mature seeds for a full year. Any mature seeds will become plants, which will then grow and develop as normal. The simulation should begin with a new seed in the middle of the field.

Evidence you need to provide: **(7 marks)**

- Your amended SOURCE CODE PROGRAM for the constants section
- Your amended SOURCE CODE PROGRAM for `CreateNewField`
- You amended SOURCE CODE PROGRAM for `SimulateSpring`
- Your amended SOURCE CODE PROGRAM for `SeedLands`
- One SCREEN CAPTURE, showing the spring output in year 1 of a two-year simulation
- One SCREEN CAPTURE, showing the spring output in year 2 of a two-year simulation

Question 14

This question refers to `CreateNewField` and the constants section.

Currently, the size of the field is fixed to a width of 35 and a length of 25. Modify the code to prompt the user to specify the length and width of the field when creating a new field. Validation should be included to ensure a minimum of 10 and a maximum of 50 for both the width and the length.

Any attempt on the part of the user to enter a value beyond this range should cause a suitable error message to be displayed.

Evidence you need to provide: **(13 marks)**

- Your amended SOURCE CODE PROGRAM for the constants section
- Your amended SOURCE CODE PROGRAM for `CreateNewField`
- One SCREEN CAPTURE showing the program's response to an attempt to give either of the field's dimensions a value of '9', having requested a one-year simulation
- One SCREEN CAPTURE showing the inputs requesting a length of 10 and a width of 20, along with spring of a one-year simulation

Question 15

This question refers to `SimulateAutumn`, and to a new function `SimulateDisease`.

The simulation is to be changed to include the possibility that a disease can strike a plant during autumn, before it has had a chance to spread its seeds. There is a 1 in 25 chance of a plant contracting this disease, which kills the plant itself and any other plants or seeds within two squares on the field. It can pass through rocks, but the rocks are in no way affected themselves.

The effects of the disease are shown below, with the disease striking the middle plant:

Before disease

P	S	•	•	P
S	S	•	X	•
•	•	P	•	•
•	P	X	•	P
P	•	P	•	•

After disease

•	•	•	•	•
•	•	•	X	•
•	•	•	•	•
•	•	X	•	•
•	•	•	•	•

A plant that has been eliminated by disease will not spread any seeds.

Create a new function called `SimulateDisease`, to have the following parameters:

- A two-dimensional character array, `Field`
- An integer, `Row`, indicating the row of the diseased plant
- An integer, `Column`, indicating the column of the diseased plant

`SimulateDisease` should return a two-dimensional character array to `SimulateAutumn`.

The program should not attempt to spread the disease outside the bounds of the field, so if the diseased plant is in the leftmost column, the disease will not spread to the left.

If disease does strike, the program should output the text "Disease has struck!" once for each plant struck by disease.

Evidence you need to provide: **(12 marks)**

- Your amended SOURCE CODE PROGRAM for `SimulateAutumn`
- The SOURCE CODE for a new function, in full, called `SimulateDisease`

