THE UNIVERSITY OF WAIKATO
Te Whare Wānanga o Waikato

2025 SCHOLARSHIP EXAMINATION

DEPARTMENT	Computer Science
COURSE TITLE	Year 13 Scholarship
TIME ALLOWED	FIVE hours with a break for lunch at the discretion of the supervisor
QUESTIONS	There are TWO questions in the paper. Candidates are to answer BOTH questions. Answer as much of each question as you can. Note that Question 2 is significantly more difficult than Question 1. Plan your time to allow a good attempt at each.
INSTRUCTIONS	Candidates may use any text or manual or online programming language documentation for reference during the examination. Candidates may not copy code from the internet or consult anyone other than the examiners during the examination
DETAILS	Both questions pose problems which you are asked to solve by writing computer programs. They may also ask for written answers for some problem parts. In programming you may work in the programming language of your choice. However, the examiners need to be able to read your program text and if at all possible, test run it. If problems arise from your choice of programming language, we may contact you after the examination, for clarification. Written answers to parts of questions can be submitted in text files; included as comments in your program text; or as photographed or scanned images of hand written documents. Remember also that partial marks may be awarded for programming ideas written down, but not yet implemented.
CALCULATORS PERMITTED	Yes

1. **Simple Weather Reporter** (Careful and Accurate Programming)

Your programming work in this question will be assessed on three criteria:

(a) Completeness and accuracy of the program. It may be that this problem statement does not state exactly what the program should do under all circumstances. If you find a situation of that nature, choose a solution and write down, either on paper or in the comments of your program what the difficulty was and how you chose to resolve it.

(b) Good presentation. That is, it should make good use of programming language facilities, be well organised, neatly laid out, and lightly commented.

(c) Careful checking. Wherever possible check input from the program user in case they have made errors.

In this question, you are asked to write a program that helps a user record and analyze the temperature for a number of days.

Your program should begin by asking the user how many days of temperature data they would like to enter. The user should then be prompted to enter one temperature reading per day. Each temperature reading should be an integer (it may not include decimal points).

Once all the temperature readings have been entered, your program should perform the following tasks:

- ⇨ Display all the temperatures that were entered.
- ⇨ Calculate and display the average temperature.
- ⇨ Determine and display the highest and lowest temperature values.
- ⇨ Count and display the number of hot days (defined as days with a temperature of 20°C or higher).
- ⇨ Count and display the number of cold days (defined as days with a temperature of 10°C or lower).
- ⇨ If the average temperature is 20°C or higher, display the message: "It was a warm week!"
- ⇨ If the average temperature is 10°C or lower, display the message: " It was a chilly week!"
- ⇨ Otherwise, display the message: "It was a mild week!"

The transcript of a sample interaction with such a program is given on the next page. In the transcript, information entered by the user is shown in **bold** type. You don't have to follow this style of data entry or format results in the same way. The sample is just here to show the kind of interaction expected of your program.

QUESTION 1 CONTINUES ON THE NEXT PAGE

```
Welcome to the Weather App!

How many days of temperature data would you like to enter? 7

Please enter the temperature for day 1: 8
Please enter the temperature for day 2: 16
Please enter the temperature for day 3: 21
Please enter the temperature for day 4: 25
Please enter the temperature for day 5: 14
Please enter the temperature for day 6: 10
Please enter the temperature for day 7: 7

Thanks! The temperatures that you entered are as follows: 8, 16, 21, 25, 14, 10, 7

The average temperature is 14 degrees Celsius
The lowest temperature is 7 degrees Celsius
The highest temperature is 25 degrees Celsius
There were 2 hot days
There were 3 cold days

It was a mild week!
```

2. **Fire Spread Simulation: Saving the Forest** (Problem Solving and Programming)

Your programming work in this question will be assessed on three criteria:

(a) Your approach to the problem. We will be looking at your work for evidence that you found good ways of storing the necessary data, and devised algorithms for finding and displaying the requested results. ***Please hand in any notes and diagrams which describe what you are attempting to program, even if you don't have time to code or complete it. You may include comments in your program, or write a description of your program to hand in.***

(b) The extent to which your program works and correctly solves the problem.

(c) The extent to which you use results from your programming to explore the problem presented.

You may find that the programming language you use makes it difficult to produce output as shown in the example implementation steps below. If this is the case, feel free to build your program in a way that suits your circumstances.

Note: Four text files have been provided to you: forest_grid_1.txt, forest_grid_2.txt, forest_grid_3.txt, and forest_grid_bad.txt.

You are an environmental scientist tracking a wildfire spreading through a forest. Your mission is to simulate the spread of fire across a grid-based map of the forest and find strategies to slow or stop the blaze.

The forest is represented as a two-dimensional grid, where each cell in the forest grid can contain a tree ('T'), empty ground ('.'), fire ('F'), or water ('W'). The fire spreads each time step from a burning tree ('F') to adjacent trees ('T') in the four cardinal directions (up, down, left, right). Water blocks the fire.

In this question, you are asked to write a program to simulate the fire spreading, and mitigate the spread of fire, using only text display. The question presents the problem in stages for you to program. We suggest that you build your program in the order given. This will make it likely that you have parts working at the end, even if you don't have time to complete the whole program. We also strongly suggest that you read through all the stages before starting to program. Stages I, J, and K are the final stages, in which you have the most freedom to explore algorithm ideas.

The stages of this problem involve building and changing a program. Instructions will be given in some detail for the first stages. Later stages require that you develop the code yourself. When you are making a major change, you should save a working version of your program. This will help us see what you have achieved, especially if you have difficulties with the altered version. Where stages ask you to try different ways of displaying the game, you can write different display procedures within the same program to make sure that all of your answers are still visible to the examiner.

Setting up the Grid

Stage A: Building a basic grid

The forest is displayed in a grid. Write a program to draw a 4 x 4 grid using text characters. The result should be similar to the figure below (each cell in the grid is one line high and three characters wide).

```
+------------+
| .  .  .  . |
| .  .  .  . |
| .  .  .  . |
| .  .  .  . |
+------------+
```

Stage B: Adding trees to the grid

The next thing you need to do is add trees to the forest. Given a 4 x 4 grid, add the following trees at the following positions:

- Trees
  - X = 1, Y = 1, symbol = T
  - X = 1, Y = 2, symbol = T
  - X = 2, Y = 2, symbol = T

The results should be similar to the figure below.

```
+------------+
| .  .  .  . |
| .  T  T  . |
| .  .  T  . |
| .  .  .  . |
+------------+
```

Stage C: Building a dynamic grid

So far, the forest is displayed in a 4x4 grid. However, forests come in all shapes and sizes Extend your program to draw a dynamic grid, based on user input. If the user entered a width of 6 and height of 4, then the result should be similar to the figure below (each cell in the grid is one line high and three characters wide).

```
+------------------+
| .  .  .  .  .  . |
| .  T  T  .  .  . |
| .  .  T  .  .  . |
| .  .  .  .  .  . |
+------------------+
```

Stage D: Reading the forest from file

Extend your program to read forest information from file. *Note: four text files have been provided to you for this section: forest_grid_1.txt, forest_grid_2.txt, forest_grid_3.txt, and forest_grid_bad.txt.*

The first line of the file represents the height and width of the grid. Each line following represents one row of the grid, where 'T' represents a tree and '.' represents empty ground. Each value is separated by a space. The first three lines of *forest_grid_1.txt* are as follows.

```
4 4
T . . .
T T . .
```

The images below show the expected output from *forest_grid_1.txt, forest_grid_2.txt,* and *forest_grid_3.txt. forest_grid_bad.txt* can be used as an example of a bad input file. Your program should handle bad input files gracefully.

```
+------------+
| T  .  .  . |
| T  T  .  . |
| .  T  .  . |
| .  T  T  . |
+------------+
```
forest_grid_1.txt

```
+----------------------+
| T  T  T  T  T  T  .  |
| T  .  .  .  .  T  .  |
| T  .  .  .  .  T  T  |
| T  T  T  T  .  .  T  |
| .  .  .  T  T  .  T  |
| .  .  .  T  .  .  T  |
| .  .  .  .  .  .  .  |
+----------------------+
```
forest_grid_2.txt

```
+---------------------+
| T  T  T  T  T  T  T |
| T  .  .  T  .  .  T |
| T  T  T  .  T  T  T |
| T  .  .  T  .  .  T |
| T  T  T  T  T  T  T |
+---------------------+
```
forest_grid_3.txt

Simulating the fire

Stage E: Ignite the fire

The next step is to specify where the fire begins. Ask the user for the coordinates of where to start the fire. Change that cell to 'F', and display the updated forest grid.

Please note, the fire can only start in a cell that contains a tree.

The output below shows an example of gameplay.

```
Welcome to the Fire Spread Simulation!

You are an environmental scientist tracking a wildfire spreading through a forest. Your
     mission is to simulate the spread of fire across a grid-based map of the forest and find
     strategies to slow or stop the blaze.

Press 1 to load a default 4x4 forest grid, press 2 to load a forest grid from file: 2

Please enter the name of the forest file: forest_grid_1.txt

Thank you, we are loading your forest grid now. Press enter to display the grid [enter]

+------------+
| T  .  .  . |
| T  T  .  . |
| .  T  .  . |
| .  T  T  . |
+------------+

It's time to start the fire simulation.
Please enter the x co-ordinate of the fire: 0
Please enter the y co-ordinate of the fire: 3
```

```
Whoops, you can only start the fire simulation in a cell that contains a tree. Please try
    again.

Please enter the x co-ordinate of the fire: 0
Please enter the y co-ordinate of the fire: 0

Thank you, we are loading your forest grid now. Press enter to display the grid [enter]

+------------+
| F  .  .  . |
| T  T  .  . |
| .  T  .  . |
| .  T  T  . |
+------------+
```

Stage F: Simulate Fire Spread

It is now time to simulate the spread of the fire. The fire spreads one step at a time. It spreads from a burning tree ('F') to adjacent trees ('T') in the four cardinal directions (up, down, left, right).

You should:

- ⇨ simulate one time step where fire spreads to adjacent trees
- ⇨ display the grid after each step
- ⇨ repeat for several steps or until fire no longer spreads

The output below shows an example of gameplay.

```
Welcome to the Fire Spread Simulation!

You are an environmental scientist tracking a wildfire spreading through a forest. Your
    mission is to simulate the spread of fire across a grid-based map of the forest and find
    strategies to slow or stop the blaze.

Press 1 to load a default 4x4 forest grid, press 2 to load a forest grid from file: 2

Please enter the name of the forest file: forest_grid_1.txt

Thank you, we are loading your forest grid now. Press enter to display the grid [enter]

+------------+
| T  .  .  . |
| T  T  .  . |
| .  T  .  . |
| .  T  T  . |
+------------+

It's time to start the fire simulation.
Please enter the x co-ordinate of the fire: 0
Please enter the y co-ordinate of the fire: 0

Thank you, we are loading your forest grid now. Press enter to display the grid [enter]

+------------+
| F  .  .  . |
| T  T  .  . |
| .  T  .  . |
| .  T  T  . |
+------------+

The fire spread simulation has now started. Press enter to see the next time step. [enter]
```

```
+------------+
| F  .  .  . |
| F  T  .  . |
| .  T  .  . |
| .  T  T  . |
+------------+

Press enter to see the next time step. [enter]

+------------+
| F  .  .  . |
| F  F  .  . |
| .  T  .  . |
| .  T  T  . |
+------------+

Press enter to see the next time step. [enter]

+------------+
| F  .  .  . |
| F  F  .  . |
| .  F  .  . |
| .  T  T  . |
+------------+

Press enter to see the next time step. [enter]

+------------+
| F  .  .  . |
| F  F  .  . |
| .  F  .  . |
| .  F  T  . |
+------------+

Press enter to see the next time step. [enter]

+------------+
| F  .  .  . |
| F  F  .  . |
| .  F  .  . |
| .  F  F  . |
+------------+

Press enter to see the next time step. [enter]

The fire spread has reached its end. The fire cannot spread any further.
```

Stage G: Save the Simulation State

As a scientist, you may want to re-analyse each step in the fire simulator at a later date. After each time step, write the current grid state to a file (e.g., fire_log.txt) to track the fire's progress.

Stage H: Analyze Burned Area

After the fire stops spreading, count: how many trees were burned, how many were saved, the total time steps taken.

Advanced gaming

Stage I, J, and K are the final stages, in which you have the most freedom to explore algorithm ideas. These stages can be completed in any order, and can be completed on their own, or in conjunction. Please note that students' completion of these stages often

contributes to their overall success in completing this question. As such, we would recommend attempting as many as you have time for.

Stage I: Add Wind Direction

At the start of the simulation, ask the user for wind direction (N, S, E, W). Simulate the fire spreading more quickly in that direction (e.g., two cells instead of one).

Stage J: Firebreaks and Strategy

As mentioned in the introduction, water ('W') blocks fire. Allow the user to replace tree tiles with water tiles before the simulation starts. Explore whether strategic placement can contain the fire.

Stage K: Animate the Fire

Use delays and repeated printing to create a simple animation of the fire spreading (without the user having to press enter between each time step).