codesprint

NATIONAL CODING COMPETITION

Finals Task

Post-Secondary Category 2026

POWERED BY mdia MALTA DIGITAL INNOVATION AUTHORITY

GOVERNMENT OF MALTA
MINISTRY FOR EDUCATION AND SPORTS
DIRECTORATE FOR STEM AND VET PROGRAMMES

× icecampus

Final Round Schedule

- Task Analysis (15 min)
- Task Development (90 min)
- Interval (15 min)
- Continuation of Task Development (90 min)
- Task Submission (15 min)
- **Top 3 Announced during the Award Ceremony**

GreenThumb Urban Garden

The GreenThumb initiative is on a mission to weave nature back into the heart of Malta's historic towns and modern urban centres. The aim is to transform rooftops, balconies, and forgotten corners into green spaces to enhance biodiversity and improve the quality of life across the islands. However, cultivating life in a Mediterranean climate requires more than just passion. It demands precision, foresight, and the careful management of precious resources like water.

As a lead developer, your goal is to build a Dynamic Simulation Dashboard that tracks predefined urban plots. Unlike standard apps, this is an autonomous simulation. Once the data is loaded, the system evolves based on the passage of time (weeks), seasonal shifts, and environmental crises. You must manage a hybrid water supply system and calculate production.

Stay sharp, think carefully, and code bravely.

Functionality #1: Program initialisation

Upon program initialisation, it should display a main menu with the below options:

1. Start a new simulation,
2. Load an existing simulation (*see functionality 7 for more details*)
3. Exit program.

⚠ **The program will only stop if the user chooses the 'Exit Program' option from the main menu.**

Functionality #2: Start a new simulation

- Upon starting a new simulation, the program should ask the user to enter the year of the simulation (e.g. 2026) and check if it already exists (*saved from previous simulations*).
- If the year already exist, display an appropriate message.
- If the year is not saved, a new simulation is started as per below:
  - start at week 1 (Winter)
  - initialise the water reserve to 1000Ltrs
  - start at 0.0kg seasonal growth
  - show the dashboard as per functionality 3 described below.
  - load the 10 reference records listed in Table 1 below that represents the data for the dashboard simulation. *The records can be either hardcoded or loaded from a file.*
  - randomly assign **ONE** compatible crop for all plots for the winter season, adhering to the crop compatibility matrix in Table 2. *Check section 3.3 for more details.*

Plot ID	Location	Plot Name	Plot Type	Water Access
1	Floriana	Argotti Courtyard	Raised Bed	Yes
2	Valletta	Bastion Lookout	Balcony	No
3	Mosta	Wied il-Għasel Edge	Raised Bed	Yes
4	Mellieħa	Parish Square Roof	Rooftop	No
5	Attard	San Anton Secret	Raised Bed	Yes
6	Victoria	Citadel Ramparts	Raised Bed	Yes
7	Xewkija	Old Mill Balcony	Balcony	No
8	Birżebbuġa	Fishermen's Wharf	Rooftop	Yes
9	Dingli	Buskett Canopy	Rooftop	No
10	Birgu	Vittoriosa Moat	Raised Bed	Yes

Table 1: Reference records

Functionality #3: The Weekly Update Engine

The program employs 'tick' logic, which allows the simulation to advance by one week each time the user requests it. You need to process the logic in the following specific order:

3.1: Hybrid Water Logic and Resource Management

The program should handle water consumption for different plot types.

- **First calculate the water needed, one plot at a time, for plots that do NOT have access to the main water supply.** These plots must always take 100% of the weekly water demand from the water reserve.
  - If the water reserve drops below 150 Ltrs, it is reset to 1000 Ltrs before the water requirements for the remaining plots will continue to be calculated. In this case, a 'BOWSER REFILL' alert is triggered for the user to see in the dashboard update.

    💡 **For example**:
    - Week 17 (Spring)
    - Water reserve available: 158 Ltrs
    - Water required for Basil planted in Plot 2 → 10 Ltrs
    - Water reserve remaining → 148 Ltrs
    - Bowser Refill (Water reserve reset) → 1000 Ltrs
    - Continue water calculations for remaining plots

- **Second calculate the water requirement, one plot at a time, for plots that have access to the main water supply.** For every plot with water access to the main water supply, the program should:
  - Use 50% of the weekly water demand from the water reserve and the remaining 50% from the main water supply.
  - If the water reserve falls below 250 Ltrs, the program must take 100% of the weekly water demand from the main water supply.

  💡 **Example 1:** when water reserve is sufficient:
  - Week 6 (Winter)
  - Water reserve available: 400 Ltrs **(> 250 Ltrs)**
  - Water required for Eggplant planted in plot 5 → 16 Ltrs
  - 50% from reserve → 8 Ltrs
  - 50% from main supply → 8 Ltrs
  - Water Reserve available for plot 6 → 392 Ltrs

💡 **Example 2:** when water reserve is low:

- Week 8 (Winter)
- Water reserve available: 216 Ltrs **(< 250 Ltrs)**
- Water required for Peppers planted in Plot 7 → 14 Ltrs
- 0% from reserve → 0 Ltrs
- 100% from main supply → 14 Ltrs
- Water Reserve available for plot 8 → 216 Ltrs

⚠ **A bowser refill is required only when calculating water demand for plots that have no access to the main water supply.**

3.2: Growth Calculation and the Yield Formula

- Growth calculation should happen at the end of every season. It is assumed that every season has 13 weeks. Therefore, growth calculation is done in Weeks 13, 26, 39, 52.
- The formula to use is `Yield = Growth Factor * Multiplier`.
- The Growth Factor for each crop is listed in Table 2 below, and the Multiplier is:
  - 0.8 for a Balcony,
  - 1.2 for a Rooftop, and
  - 2.2 for a Raised Bed.

💡 **Example Calculations for Week 13 (End of Winter):**

- Plot with a Raised Bed growing Peas: **Yield** = `8.5 * 2.2 => 18.70Kg`
- Plot with a Rooftop growing Eggplant: **Yield** = `10.5 * 1.2 => 12.60Kg`
- Plot with a Balcony growing Spinach: **Yield** = `5.5 * 0.8 => 4.40Kg`

3.3: Auto-Seasonal Rotation

When a new season starts (weeks 1, 14, 27, and 40), the program must randomly assign a new crop to each plot.

⚠ It must only choose from crops 1) pertaining to the current season, and 2) compatible with that specific plot type as per the crop compatibility matrix shown in Table 2 below.

✅ All crops must be planted at least once.

💡 **For example:**

- A balcony will never be assigned an Eggplant.
- A raised bed will never be assigned a Basil.

Week Starting	Season	Crop Name	Compatible Plot Types	Weekly Water Demand (Ltrs)	Growth Factor
1	**Winter**	Peas	Rooftop, Raised Bed	12	8.5
		Strawberries	Balcony, Rooftop, Raised Bed	8	6.2
		Spinach	Balcony, Rooftop, Raised Bed	7	5.5
14	**Spring**	Basil	Balcony, Rooftop	10	4.5
		Peppers	Balcony, Rooftop, Raised Bed	14	9.8
		Green Beans	Rooftop, Raised Bed	11	8.0
27	**Summer**	Zucchini	Rooftop, Raised Bed	20	12.5
		Cherry Tomatoes	Rooftop, Raised Bed	18	11.0
		Eggplant	Rooftop, Raised Bed	16	10.5
40	**Autumn**	Celery	Balcony, Raised Bed	10	7.4
		Rucola	Balcony, Rooftop, Raised Bed	6	3.8
		Radish	Balcony, Rooftop, Raised Bed	5	4.2

Table 2: Crop Compatibility Matrix

📢 **Important Note:**

For convenience, a Word document and a csv file containing the list of plots and crops is available on https://bit.ly/cs26psec_greenthumb.

3.4: The Environmental Crisis Generator

- Every week there is a 25% chance of a random environmental event occurring as per Table 3 below.
- The program must check for a random event before calculating water consumption.
- In case that an environmental event happens, a system alert is triggered in the dashboard update.

Event	Season	Impact
Heatwave	Summer	All plots currently in the simulation lose 1.5 Kg of their seasonal yield. (Yield cannot drop below 0kg).
Gregale Storm	Winter & Autumn	Add 300 Ltrs to the Water Reserve (capped at 1000 Ltrs). However, the wind is destructive, and one random plot will have its crop destroyed (0 Kg seasonal yield).
Pest Outbreak	Spring	A specific pest targets a crop. The program must pick one specific crop type (e.g., Strawberries) currently being grown. Every plot growing that specific crop loses 40% of its seasonal yield.

Table 3: Environmental Crisis Event List

Functionality 4: Dashboard Display Requirements

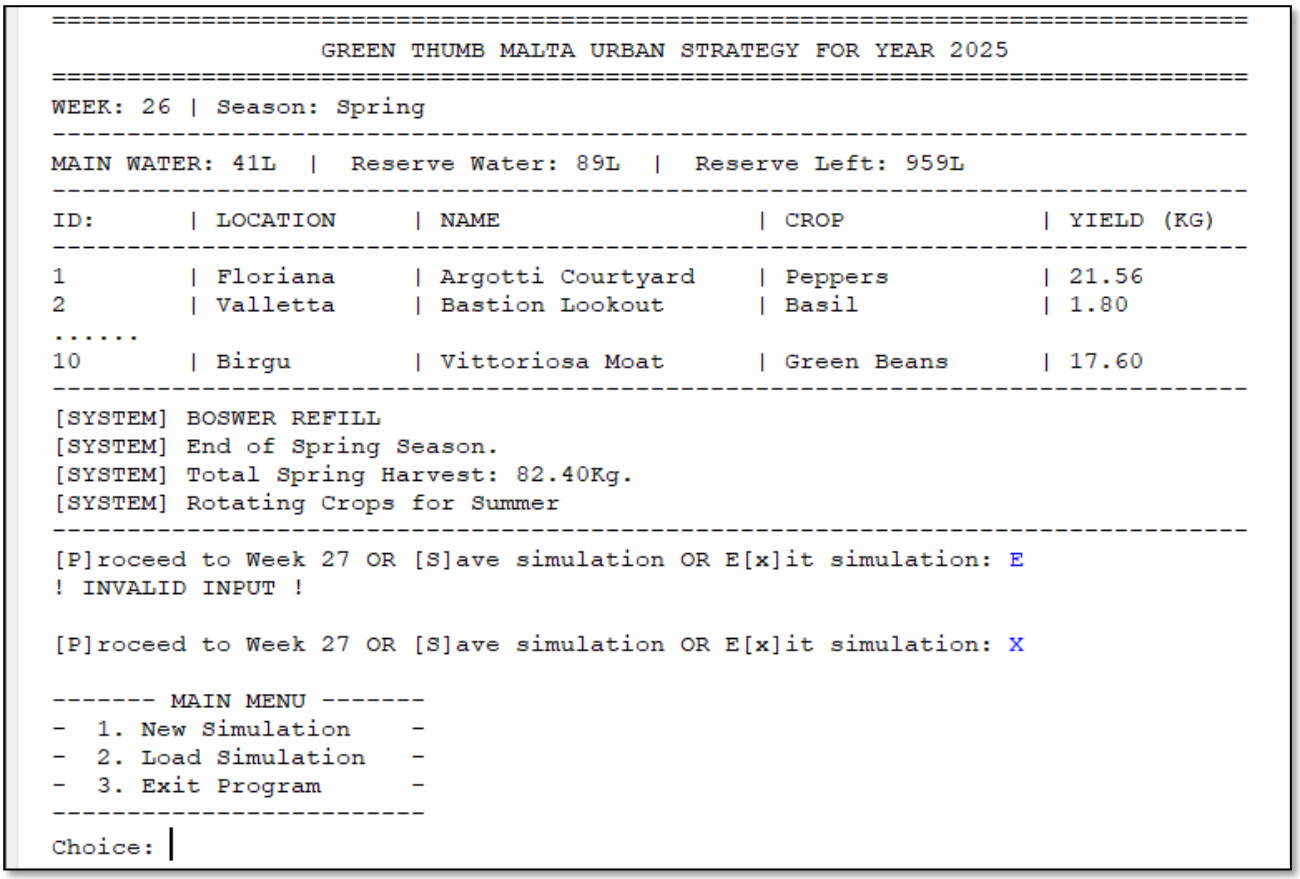
The simulation should display a user-friendly dashboard every week. It could be text-based or graphics-based. The dashboard must include:

- A clear title displaying "GREEN THUMB: MALTA URBAN STRATEGY".
- A status bar that shows the current week number and the corresponding season, the main water supply usage in litres, the reserve water usage in litres, and the current water reserve level in litres.
- A formatted table displaying all 10 plots. The columns must include ID, Location, Plot Name, and Crop. The table should also include another column, named Yield (Kg), every 13th week (end of season) that displays the total production growth per plot.
- Clear system notification updates below the table indicating any event that happened during that particular week, such as a bowser refill, a heatwave, etc...
- A final prompt line instructing the user to 1) Press ENTER to advance to next week, or 2) save simulation, or 3) exit simulation.
- If the 'save simulation' option is chosen (check functionality 5 for more details), the current details of the simulation should be saved, and the user re-prompted to 1) advance to next week, or 2) exit the simulation.

Text-based sample visuals are shown in Screenshot 1 and Screenshot 2 below.

```
================================================================================
               GREEN THUMB MALTA URBAN STRATEGY FOR YEAR 2025
================================================================================
WEEK: 2   | Season: Winter
--------------------------------------------------------------------------------
MAIN WATER: 34L   |  Reserve Water: 76L   |  Reserve Left: 1000L
--------------------------------------------------------------------------------
ID:       | Location      | Name                  | Crop
--------------------------------------------------------------------------------
1         | Floriana      | Argotti Courtyard     | Peas
2         | Valletta      | Bastion Lookout       | Spinach
......
10        | Birgu         | Vittoriosa Moat       | Strawberries
--------------------------------------------------------------------------------
[SYSTEM] Greagle Storm Hit the Island
[SYSTEM] Reserve Water filled to 1000L
[SYSTEM] Strawberries of Plot MLT-007 were destroyed
--------------------------------------------------------------------------------
[P]roceed to Week 3 OR [S]ave simulation OR E[x]it simulation: s
 Simulation saved ...
[P]roceed to Week 3 OR E[x]it simulation:
```

Screenshot 1: Sample dashboard visual for Week 2 (Winter)



Screenshot 2: Sample dashboard visual for week 26 (end of Spring)

Functionality 5: Final Year-End Summary (Week 52)

- After week 52, the user can only proceed to the Annual Performance Summary Report. The user should not be given the option to save or exit the simulation.
- The Annual Performance Summary Report should contain the:
  - Total annual harvest (in Kg).
  - Total water consumed from reserve (in Ltrs).
  - Total water consumed from main supply (in Ltrs).
  - Total bowser refills.
  - Water Efficiency Ratio (WER) following the formula:

    WER = Total Water Consumed (Reserve + Supply) / Total Annual Harvest (Kg)

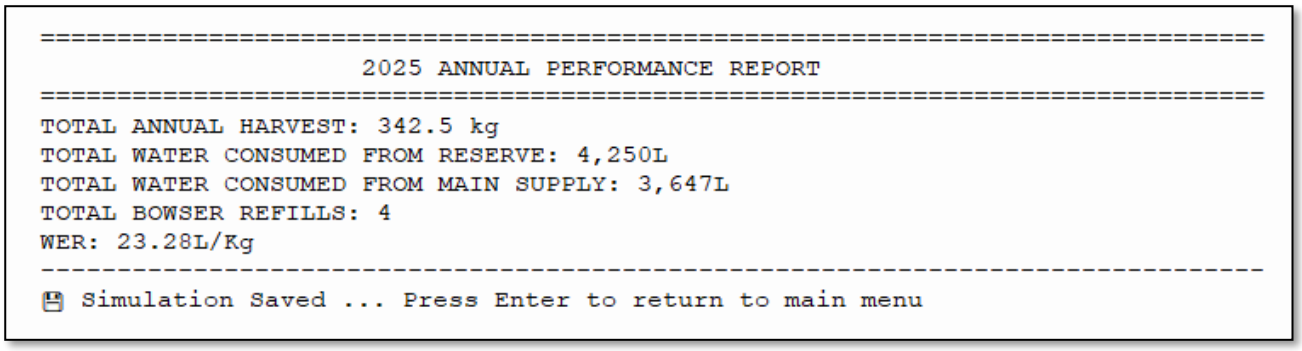
    💡 **For example,**

    Total Water Consumed (4325 + 3647) = 7972 Ltrs

    Total Annual Harvest = 342.5 Kg

    WER (7972 / 342.5) = 23.28 Ltrs / Kg
- Upon displaying the Annual Performance Summary Report, the program should automatically save the simulation as per functionality 6.

💡 **A text-based sample visual is shown in Screenshot 3.**



Screenshot 3: Sample visual of the Annual Performance Summary Report

Functionality #6: Save Simulation

When the user chooses to save the simulation, or it is being saved automatically at the end of the year, the program should:

- Save the current state of the simulation in one, or more than one file.

  It is the discretion of the participant to choose how to save the records. The file, or files, can be of any type, such as text file, json file, csv file etc.

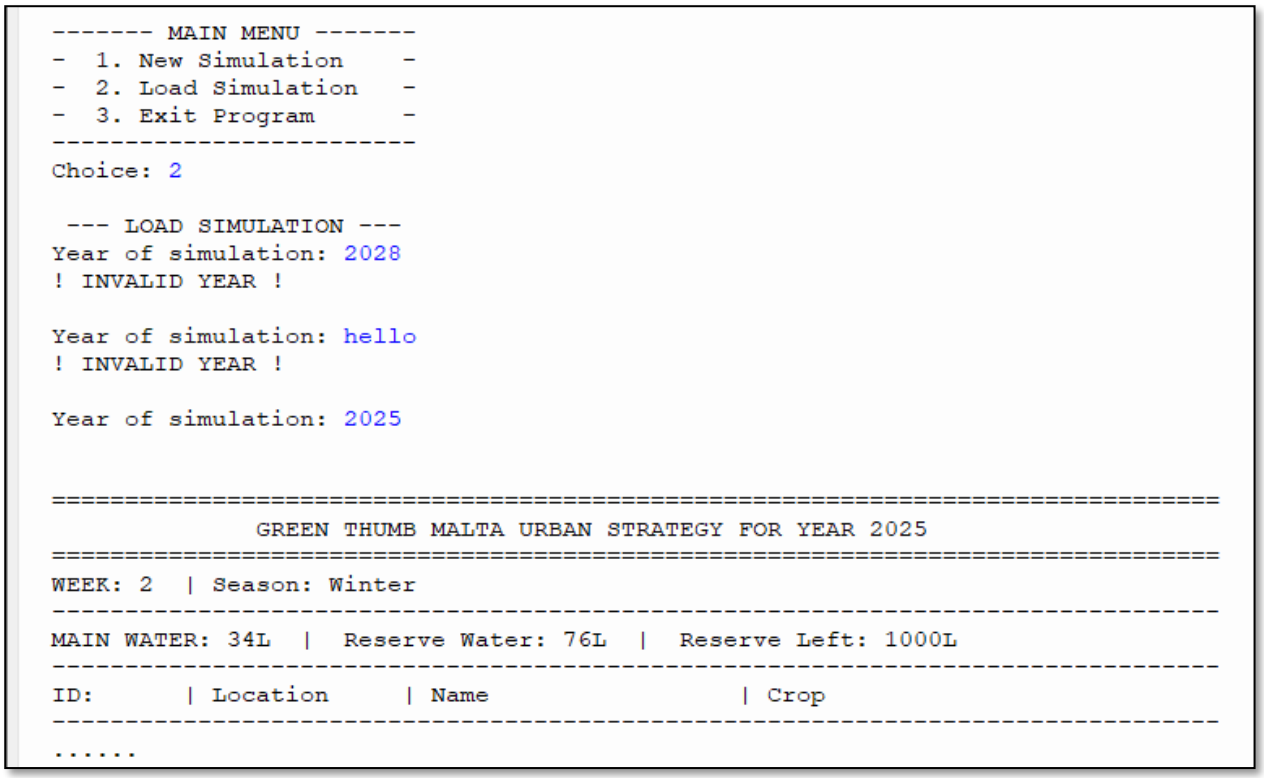
- The following details should be saved:
  - Year,
  - Current Week number,
  - Current Water Reserve level,
  - Cumulative Reserve Water used,
  - Cumulative Main Water Supply used,
  - Cumulative Seasonal Crop Yield,
  - Total count of Bowser Refills,
  - For every Plot save the following details: ID, currently assigned Crop.

Functionality #7: Load Simulation

If the 'Load a simulation' option is chosen, the program should prompt the user to enter the year of the simulation.

- If the records of that year exist:
  - display the Annual Performance Report if the year has ended, OR
  - reload all the details and resume execution from the exact week, season, water state and total crop yield.
- If the records of that year do not exist, it should display an appropriate message.

Text-based sample visuals are shown in Screenshot 4.



Screenshot 4: Loading a simulation visual sample

Functionality #8: Check Routines

To maintain simulation stability, your code must prevent any runtime errors and implement the following checks:

- User input is NOT case sensitive.
- The year should be between 2020 and 2026.
- For any invalid input, an appropriate prompt should be displayed.
- The water reserve must never be a negative value. If a calculation would result in a negative number, a bowser refill should apply before the water calculation proceeds.
- Seasonal yield for any plot must never drop below 0 Kg, especially after a 'Maltese heatwave' crisis.
- During seasonal rotation, the program must check that the randomly selected crop is compatible with the plot type (e.g., zucchini cannot be assigned to a balcony).

Functionality #9: Modular Programming

- Structure your code modularly, using any preferred programming paradigm of your choice, being it function base and/or object-oriented.
- Use any data structure/s of your choice where appropriate.
- Include comments and apply Java or Python naming conventions.

FOR PYTHON:

Name your program **main_app.py**,
include all files into a folder called **postSec_greenThumb**,
*and submit as a Zip file named **postSec_name_surname**.*

FOR JAVA:

Name the class containing the main method **'RunApp'**.
include all files into a folder called **postSec_greenThumb**,
*and submit as a Zip file named **postSec_name_surname**.*

Assessment Rubric

Core Functionality (32 points)	
Program Initialisation (Main Menu)	Show all options and program ends only when user choses the exit option from the main menu.
New Simulation Initialization	Correctly initializes the simulation state (Week 1, 1000 Ltrs water reserve, 0 Kg seasonal growth), loads all plots records, and plant crops appropriately.
Weekly Update Engine	Implements the 'tick' logic to advance the simulation by one week upon user request and process the simulation logic in the specified order.
Water Logic - No Main Access	Accurately calculates and deducts 100% of weekly water demand from the water reserve for plots without main water access.
Bowser Refill Mechanism	Correctly triggers a 'BOWSER REFILL' alert and resets the water reserve to 1000 Ltrs when it drops below 150 Ltrs, specifically for plots without main water access.
Water Logic - With Main Access	Accurately calculates water consumption for plots with access to main water supply, using 50% from reserve and 50% from main supply, or 100% from main supply if reserve is below 250 Ltrs.
Seasonal Growth Calculation	Correctly applies the Yield = Growth Factor * Multiplier formula at the end of each season (Weeks 13, 26, 39, 52) for all plots.
Auto-Seasonal Crop Rotation	Automatically updates crops for all plots at the start of each season (Weeks 1, 14, 27, 40) by randomly assigning new crops.
Crop Compatibility	Randomly assigned crops are strictly compatible with the plot type as per the provided crop compatibility matrix and season. Also, all crops should be planted at least once.
Environmental Crisis Handling	Implements the 25% weekly chance of an environmental event and correctly applies the specified impacts for Heatwave, Gregale Storm, and Pest Outbreak.
Year-End Summary Report	Displays the Annual Performance Summary Report after Week 52, including total annual harvest, total water consumed from reserve and main supply, and total bowser refills.
Input Requirements	Prompts user for all input required (main menu options, year input, and options chosen during simulation)
Arithmetic Precision	All calculations, including water consumption, growth, and yield, are performed with correct arithmetic and precision.
Save Simulation	Correctly saves all simulation data (Year, Week, Reserve, Cumulative Water metrics, Bowser Refills, Plot ID, and current crop) into one or more files.

Load Simulation	Successfully loads saved year configurations from file/s. Automatically displays the Annual Summary if the year has ended or restores all details and resumes active simulation execution.
Other Functionalities	Other possible 'hidden' functionalities work correctly.
Validation (6 Points)	
Input & Case-Insensitive Validation	All input handlers are case-insensitive, and the program can handle invalid keyboard entries.
Value Constraint Enforcement	Implements validation checks so water reserve and seasonal growth never fall below 0.
Avoidance of Run Time Errors	Program does not crash unexpectedly during runtime but traps user input errors, corrupt save files, and missing assets without crashing the execution environment.
Technical Implementation (6 Points)	
Use of Data Structures	Uses appropriate Python data structures (such as nested lists, dictionaries, or custom objects) to structure plot states and crop details.
Modular Programming	Code is well-structured into modules through user-defined functions and/or object-oriented programming.
Code Efficiency & Readability	Code is efficient, avoids unnecessary complexity, and is easy to understand.
User Experience (10 Points)	
Dashboard Display	Presents a **user-friendly** dashboard with the specified title, status bar, and formatted plots table. Both text-based and GUI are accepted!
System Notifications	Clearly displays system alerts and notifications (e.g., 'BOWSER REFILL', environmental events) below the plots table.
Java / Python Conventions	Adheres to Java / Python naming conventions (e.g., camelCase / snake_case), uses meaningful variable names, and maintains consistent indentation.
Code Comments	Includes clear and concise comments to explain complex logic, functions, and important sections of the code.
File Naming	The program file is correctly named according to the specified format
0 – Not Satisfactorily | 1- Partly Satisfactorily | 2- Entirely Satisfactorily	

Maximum Score: 54 + 2 for every extra feature*

* Extra features should extend the task functionality or demonstrate technical creativity WITHOUT altering or interfere with the core rules or required functionality of the task.

codesprint

ORGANISED BY

GOVERNMENT OF MALTA
MINISTRY FOR EDUCATION AND SPORT
DIRECTORATE FOR STEM AND VET PROGRAMMES

×

icecampus

POWERED BY TOP BRANDS

MAIN TITLE SPONSORS

mdia MALTA DIGITAL INNOVATION AUTHORITY

me direct

COMMUNITY & EXPERIENCE SPONSORS

pwc

melita

Bolt

KNIGHTS COLLEGE

JOli

XJENZA MALTA

SUPPORTING SPONSORS

mita

edQuanta RESEARCH FOR A BETTER TOMORROW

computime BUSINESS SYSTEMS ENGINEERING

9H

MEDIA PARTNERS

Lovin Malta

TIMES OF MALTA

meet INC.