code sprint MT
NATIONAL CODING COMPETITION

Final Round

Secondary Competition 2025

POWERED BY mdia MALTA DIGITAL INNOVATION AUTHORITY

GOVERNMENT OF MALTA
MINISTRY FOR EDUCATION, SPORT, YOUTH, RESEARCH AND INNOVATION

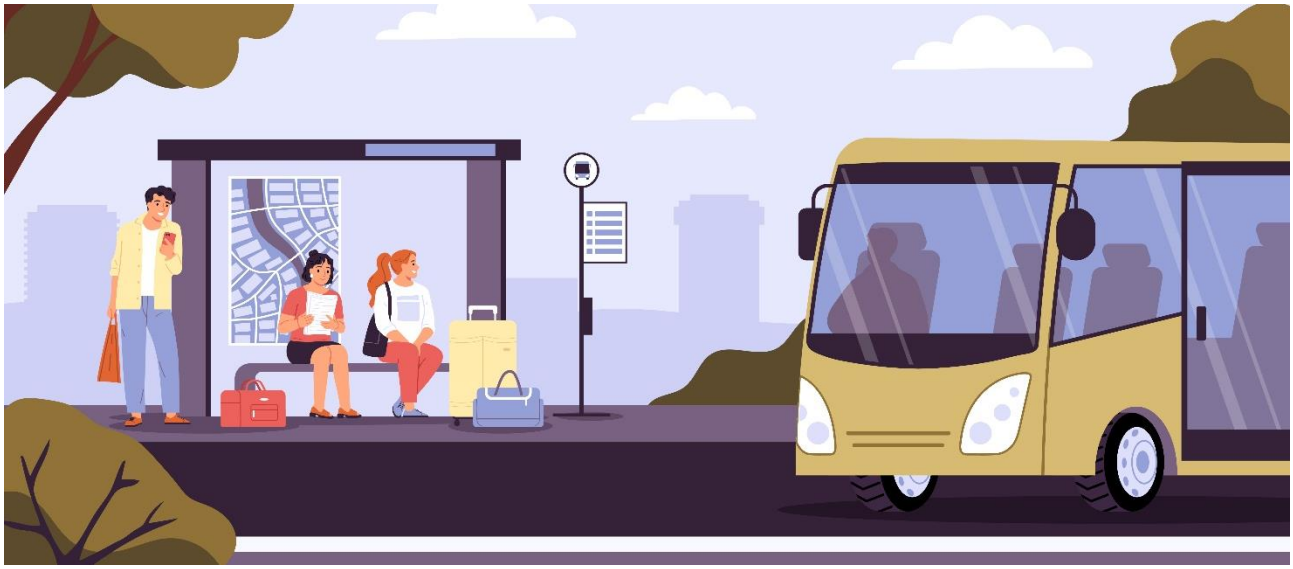
× icecampus

Final Round Schedule

- Task Analysis (10 min)
- Task Development (90 min)
- Interval (15 min)
- Continuation of Task Development (90 min)
- Task Submission (5 min)
- **Top 3 announced during the Award Ceremony**

BusIt Malta – Track – Connect - Transfer!

In the bustling island of Malta, public transport connects towns, villages, and key landmarks across the archipelago. With thousands of daily commuters, locals, and tourists relying on an efficient system, Malta Public Transport is investing in a next-generation solution, called BusIt Malta, a smart system designed to manage bus routes, bus stops, and provide transfer options between different routes across the South, Central, and Northern regions of the country.



Your challenge is to develop a simulation of the bus route scheduling software using Python and must be completed within 3 hours. Participants should focus on functionality first before optimizing or adding additional features.

Functionality #1: User Interface.

Create an interface with the following options.:

=== BusIt Menu ===
1. Add a new bus route
2. Search for bus route by route code
3. Search for bus route/s by bus stop
4. Find transfer stops between two routes
5. Show best route/s (Start ➞ Stop Journey)
6. Exit

- The program will stop only if the user chooses option 6.
- On every operation, the system should provide access to the relevant function and provide meaningful feedback or error messages.

Functionality #2: Predefined bus routes

The system should include several predefined bus routes as per the table 1 below.

📣 For convenience, a sample text file named *routes.txt* is provided and can be downloaded from www.codesprintmalta.edu.mt/routes.txt.

Bus Route	Journey (Bus stops in sequence)
R001	Triton Fountain, City Gate, Upper Barrakka, Kastilja, Konkatidral, Republic Street, Mediterranean Conference Centre, Lower Barrakka, Waterfront, Park & Ride, Triton Fountain
R003	Valletta, Floriana, Hamrun, Santa Venera, Birkirkara, Iklin, Naxxar, Mosta
R037	Valletta, Floriana, Pieta, Msida, University, Mater Dei Hospital
R041	Valletta, Floriana, Marsa, Paola, Fgura, Zabbar, Kottonera
R045	Valletta, Floriana, Msida, Birkirkara, Balzan, Attard, Ta Qali, Rabat, Mdina, Dingli
R072	Marsa, Hamrun, Qormi, Luqa, Zebbug, Siggiewi, Ghar-Lapsi
R081	Paola, Tarxien, Gudja, Luqa, Kirkop, Zurrieq, Hal Far, Birzebbuga
R084	Fgura, Bormla, Birgu, Senglea, Zabbar, Marsascala, Xghajra
R201	Luqa, Airport, Mqabba, Qrendi, Blue Grotto, Hagar Qim, Siggiewi, Rabat, Mtarfa
R212	Msida, Gzira, Sliema, St Julian, Pembroke, Swieqi, Bahar ic-Caghaq, Bugibba
R218	Bugibba, Qawra, St Pauls Bay, Xemxija, Mellieha, Cirkewwa
R223	Cirkewwa, Mellieha, Ghadira, Golden Bay, Mgarr, St Pauls Bay

Table 1: Predefined Bus Routes

Functionality #3: Add new bus route

1. This is a password protected feature, accessible only by the password `LetMe!n`
2. The interface must allow the user to add a new bus route by entering a:
   - route code, such as R101, and
   - comma-separated list of bus stops in order, such as: Valletta, Floriana, Msida, Birkirkara, Mosta.
3. The bus routes code should start with the letter 'R' and followed by a three-digit number.
4. The system should not allow bus routes with the same route code.
5. Each bus route cannot have a duplicate bus stop.

6. There cannot be more than one bus route to have the same sequence of bus stops. Bus routes should be unique! For example: **Marsa, Hamrun, Mosta**, and **Mosta, Marsa, Hamrun** are considered as two (2) unique routes.
7. Return to BusIT Main Menu if password is incorrect, or bus route is invalid or not unique.

```
=== Bus Route Manager ===
Enter password to continue: hello
Access denied. Incorrect password.
```

Screenshot 1: Sample interface for Functionality 3

```
=== Bus Route Manager ===
Enter password to continue: LetMe!n

Enter route code (e.g., R101): A104
Invalid route code format. It should start with 'R' followed by 3 digits (e.g., R105).
```

Screenshot 2: Another sample interface for Functionality 3

```
=== Bus Route Manager ===
Enter password to continue: LetMe!n

Enter route code (e.g., R101): R104
Enter comma-separated list of stops: Valletta, Floriana, Hamrun, St Venera, Birkirkara
Route R104 added successfully: Valletta → Floriana → Hamrun → St Venera → Birkirkara
```

Screenshot 3: Another sample interface for Functionality 3

Functionality #4: Search for bus route/s by route code

This feature should support:

1. Searching for a bus route and display all its stops.
2. Display an appropriate message if the bus route is not found.
3. Display an appropriate message if an invalid bus route is entered.
4. If the user types in the word '`all`', the system displays all the bus routes and their stops.

```
=== Bus Route Search ===
Enter route code (e.g., R001) or 'all' to list all routes: R333
✗ Bus route not found.
```

Screenshot 4: Sample interface for Functionality 4

```
=== Bus Route Search ===
Enter route code (e.g., R001) or 'all' to list all routes: R072

Route R072:
Ħamrun → Qormi → Marsa → Luqa → Żebbug → Siġġiewi
```

Screenshot 5: Another sample interface for Functionality 4

```
=== Bus Route Search ===
Enter route code (e.g., R001) or 'all' to list all routes: all

All Bus Routes and Stops:

R001:
Triton Fountain → City Gate → Upper Barrakka → Kastilja
→ Konkatidral → Republic Street → Mediterranean Conference Centre → Lower Barrakka
→ Waterfront → Park & Ride → Triton Fountain

R003:
Valletta → Floriana → Hamrun → Santa Venera
→ Birkirkara → Iklin → Naxxar → Mosta

R037:
Valletta → Floriana → Pieta → Msida
→ University → Mater Dei Hospital

R041:
Valletta → Floriana → Marsa → Paola
→ Fgura
```

Screenshot 6: Another sample interface for Functionality 4

Functionality #5: Search for bus route/s by bus stop

This feature should support:

1. Searching for a bus stop to identify all routes passing through it.
2. Display an appropriate message if a bus route is not found.

```
Enter a bus stop to search for: bugiba
✗ Sorry, no bus route was found serving the stop 'bugiba'.
```

Screenshot 7: Sample interface for Functionality 5

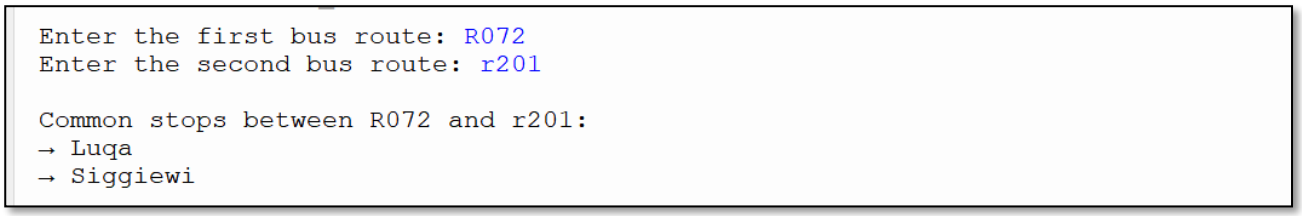
```
Enter a bus stop to search for: bugibba
Bus stop 'bugibba' is served by the following route(s):
 - R212
 - R218
```

Screenshot 8: Another sample interface for Functionality 5

Functionality #6: Find transfer stops between two routes

This feature should help users identify transfer locations by finding common stops between two routes.

1. The user should enter two bus routes, and the system displays the common stops.
2. Display an appropriate message if any one of the bus routes is not found.
3. Display an appropriate message if any one of the bus routes is invalid.
4. Display an appropriate message if bus routes are the same.



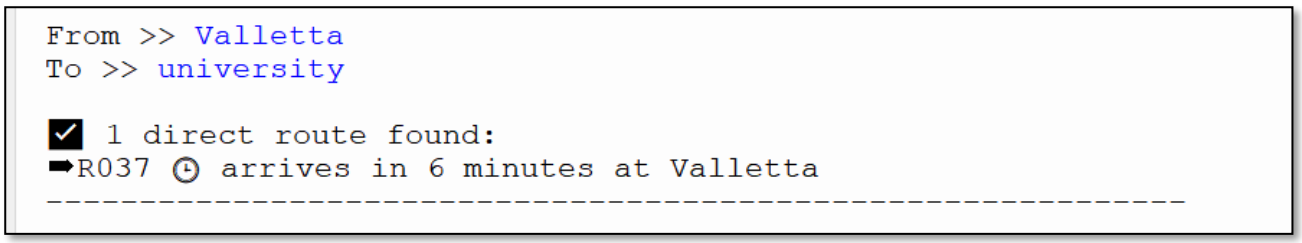
Screenshot 9: Sample interface for Functionality 6

Functionality #7: From → To Journey Planner

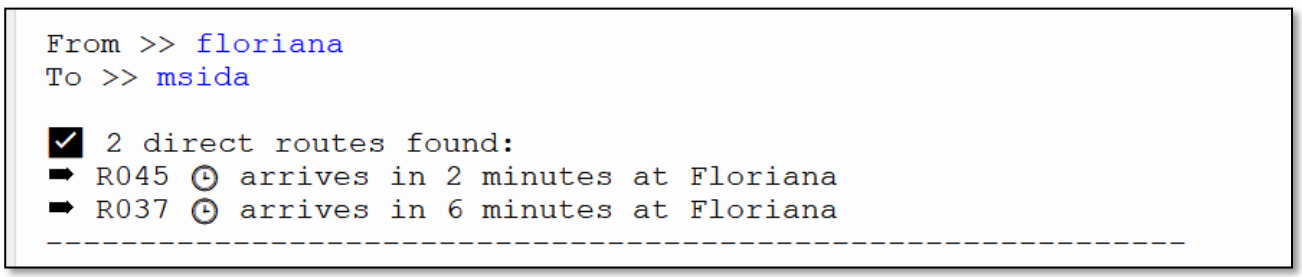
This feature helps users identify the best 'direct' bus route/s to reach their destination bus stop.

1. The user should enter the 'From' bus stop and the 'To' bus stop. The system should then display the direct bus route/s available.
2. The results must include **only direct routes**, such as route R003 from Floriana to Iklin.

   (There is no need to find routes requiring transfers, such as, from Valletta to Mellieħa which requires R037 → transfer to R212 → transfer to R218.)
3. Direct bus routes should only include journeys where the 'From' bus stop comes before the 'To' bus stop along the route. *For example, a route from Marsa to Valletta would not exist if the bus travels from Valletta to Marsa.*
4. For each suggested bus route, generate a random time between 5 and 15 minutes to simulate the arrival time at the initial bus stop.
5. If more than one direct route is available, the system should display all the options in sequence together with the arrival time at the starting bus stop.
6. Display an appropriate message if either the 'From' or 'To' bus stop is not found.
7. Display an appropriate message if direct bus routes are not found.



Screenshot 10: Sample interface for Functionality 7



Screenshot 11: Another sample interface for Functionality 7

```
From >> Marsa
To >> Birzebbuga
✗ No direct routes available
------------------------------------------------------------
```

Screenshot 12: Another sample interface for Functionality 7

Functionality #8: Validation Processes

In addition to the validation processes mentioned in the previous features, your program must implement the following validations:

1. Empty user input should be ignored and not treated as invalid. The program should re-prompt the user to enter the requested data/option.
2. All user inputs should be case-insensitive to improve usability and prevent errors due to case mismatches.
3. Prevent any runtime errors due to invalid user actions.

Functionality #9: Modular Programming

1. Structure your code using user-defined functions, such as add_route(), search_by_stop(), validate_route, generate_arrival_time, etc...
2. Use data structures, such as dictionaries, lists, tuples, and/or sets where appropriate.
3. Include comments and apply Python naming conventions.

Name your program **BusIt_name_surname.py**,
such as BusIt_john_abela.py

Submit your work as a zip file with the same name.

Assessment Rubric

Program Runs	User Friendly /Experience Interface	Proper use of Comments	Proper Conventions (Snake Case, meaningful var names etc.)	User Input
Suitable Prompts / Messages displayed	Proper Filename	Adding Bus Routes	Search Bus Route	Display 'ALL' Bus Routes
Search for Route/s by Bus Stop	Find Transfer Stops Between Two Routes	To/From Available Direct Routes	Generating Random Arrival Time	Password Protected Feature
Including the Pre-defined Bus Routes	Proper use of Data Structures	Modular Code	Code Efficiency	Other Functions / Flow
Validations				
Valid Route Codes	Route Code Already Exists (Adding new Routes)	Non-Duplicate Bus Stops in Sequence (Adding new Routes)	Unique Route Code & Bus Stops (Adding new Routes)	
Validations				
Same Route Codes (Finding Transfer Stops)	Ignore Empty Input	Avoid Runtime Errors	User Input Case Sensitivity	
0 – Not Satisfactorily | 1- Partly Satisfactorily | 2- Entirely Satisfactorily				

Maximum Score: 56 + 2 for every extra feature

code sprint MT
NATIONAL CODING COMPETITION

ORGANISED BY

GOVERNMENT OF MALTA
MINISTRY FOR EDUCATION, SPORT, YOUTH, RESEARCH AND INNOVATION

×

icecampus

POWERED BY TOP BRANDS

MAIN TITLE SPONSORS

mdia MALTA DIGITAL INNOVATION AUTHORITY

me direct

COMMUNITY & EXPERIENCE SPONSORS

melita

Bolt Business

KNIGHTS COLLEGE

pwc

JOli

SUPPORTING SPONSORS

Tech.mt
MALTA - LEADING THROUGH INNOVATION

XJENZA MALTA

cisco Networking Academy

9H

Logicom

NGU TECHNOLOGIES

MEDIA PARTNERS

Lovin Malta

TIMES OF MALTA

meet INC.