FINAL EXAM OF ALGORITHMS AND DATA STRUCTURES

Learning objectives and implementation

Objectives:

- Practical application of the techniques learned in the algorithms and data structures module of the Algorithms and Principles of Computer Science course.
- Implementation of a solution to a problem by paying attention to concrete aspects of code efficiency.

Implementation:

- C language (C11, VLA permitted).
- Only permitted library: standard C (libc).
- No multithreading.
- Input data received via stdin, results to be provided via stdout.

Evaluation Criteria

- Correctness and efficiency of the proposed solution are evaluated with automated test batteries.
- Example inputs/outputs will be provided in order to test the solution locally.

Route finder project

Motorway: list of service stations

- Each station is identified by its distance (natural number) from the beginning of the motorway.
- Each station is equipped with a set of electric vehicles with autonomy given by a positive integer.

Objective:

- Given a pair of stations, identify the route to get from the first to the second in the smallest number of stages.
- At each stop made, it is necessary to change vehicles, using one of those available at the service station.
- Attention: the route is only planned, so no cars will be moved at the request of the route.

Commands and expected responses

- add-station distance number-cars car-1 ... car-n
  - Adds a station to the motorway, identified by distance and having number-cars vehicles.
  - The autonomy of each vehicle is listed after the number of vehicles.
  - If a station already exists at the given distance, no addition is made.
  - Expected output: added / not added
- demolish-station distance
  - Removes a station already exists at the given distance.
  - It does given distance, does nothing otherwise.
  - Expected output: demolished / not demolished

Commands and expected responses

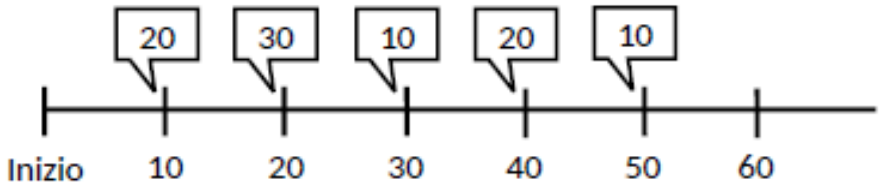
- add-car distance-station car-autonomy
  - Adds a car with the specified autonomy to the given distance station, provided the station exists.
  - N.B. It is possible to have more than one car with the same autonomy.
  - Expected output: added / not added
- scrap-car distance-station car-autonomy
  - Removes a car with the given autonomy from the station whose distance is given.
  - If the station does not exist, or a car with the indicated range does not exist in the station itself, it does nothing.
  - Expected output: scrapped / not scrapped

Commands and expected responses

- plan-route station-departure station-arrival
  - Plans the route with the least number of stages between the station of departure and the arrival station, if one exists.
  - Stations are identified in the command by their distance.
  - Expected output:
    - If the route exists: sequence of stations making up the route, in order of route, including the start and end station. Stations are printed as whole numbers, separated by spaces, on a single line.
    - If the route does not exist: no route
  - Please note: the motorway has two directions of travel.

Ambiguous paths

- There may be several equal long paths between two stations
- The implementation must choose the route that in its final part always favours the stations closest to the start of the motorway (the stations with the lowest possible number)
- This rule is not influenced by the direction of travel
- Example (in italian):



Il percorso corretto tra 10 e 60 è 10→20→40→60
Non sono corretti 10→30→40→60 perché 30 > 20
Non è corretto 10→20→50→60 perché 50 > 40

Example of input and responses expectations

Testo in ingresso (stdin)	Risposta attesa	Commento
aggiungi-stazione 20 3 5 10 15	aggiunta	Aggiunta staz. km 20
aggiungi-stazione 4 3 1 2 3	aggiunta	Aggiunta staz. km 4
aggiungi-stazione 30 0	aggiunta	Aggiunta staz. km 30
demolisci-stazione 3	non demolita	Non esiste staz. km 3
demolisci-stazione 4	demolita	Demolita staz. km 4
aggiungi-auto 30 40	aggiunta	Aggiunta auto aut. 40
aggiungi-stazione 50 3 20 25 7	aggiunta	Aggiunta staz. km 50
rottama-auto 20 8	non rottamata	Non esiste auto aut. 8
rottama-auto 9999 5	non rottamata	Non esiste staz. km 9999
rottama-auto 50 7	rottamata	Rottamata auto aut. 7
pianifica-percorso 20 30	20 30	Usa auto con aut. 15
pianifica-percorso 20 50	20 30 50	Usa auto con aut. 15 e 40
pianifica-percorso 50 30	50 30	Usa auto con aut. 25
pianifica-percorso 50 20	50 30 20	Usa auto con aut. 25 e 40
aggiungi-auto 50 30	aggiunta	Aggiunta auto aut. 30
pianifica-percorso 50 20	50 20	Usa auto con aut. 30 e 40