

# RATP App - PROJECT

*MU4RBI02— : Advanced C++*

Emir TAS

Date : 30/04/2023

Master 1 : Automation & Robotics : Smart System Engineering



## Part 1 :

### **Question 2 :**

Indeed, the file did not compile in this first instantiation phase because 'station\_hashmap' of the 'Generic\_station\_parser' class was in 'protected'. At this stage, to make the displays, I had to put them in public, otherwise it would have been necessary to put 'getters', but it was more preferable not to change too much the body of these classes which already implemented.

### **Question 4 :**

The first problem encountered was the use of the 'Grade 2019' test method. The object file did not launch during compilation, despite several checks, this part of the self-evaluation was not solved. So a simpler approach was developed.

```
When compiling including the 'object' file Grade 2019:

g++ -std=c++11 -Wall -Wextra -Werror -pedantic -pedantic-errors -O3 My_station_parser.cpp main.cpp "Grade 2019 clang-1001.0.46.3 MacOSX.o" -o main.exe

There is a problem that arises that cannot be solved :

(base) emir@MacBook-Air-de-Emir Resources-20230329 % g++ -std=c++11 -Wall -Wextra -Werror -pedantic -pedantic-errors -O3 My_station_parser.cpp main.cpp "Grade 2019
ld: warning: ignoring file Grade 2019 clang-1001.0.46.3 MacOSX.o, building for macOS-arm64 but attempting to link with file built for unknown-x86_64
Undefined symbols for architecture arm64:
  "travel::Grade::stations(travel::Generic_station_parser const&)", referenced from:
    _main in main-6ac6bb.o
  "travel::Grade::Grade(bool)", referenced from:
    __GLOBAL__sub_I_main.cpp in main-6ac6bb.o
ld: symbol(s) not found for architecture arm64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

Fig. 1 : Grade 2019 - Problem

Manual displays were made in the instantiation at this first step :

```
(base) emir@MacBook-Air-de-Emir Resources-20230329 % ./main.exe
Station: J (line 10)
Station: I (line 9)
Station: H (line 8)
Station: G (line 7)
Station: F (line 6)
Station: E (line 5)
Station: D (line 4)
Station: C (line 3)
Station: B (line 2)
Station: A (line 1)
```

Fig. 2 : 'read\_stations' display

## **Part 2 :**

### **Question 5 :**

Same approach as *question 2*, some manuel displays have been made :

```
(base) emir@MacBook-Air-de-Emir Resources-20230329 % ./main.exe
From station : 1 To station : 3 - with a period of 217 seconds.
From station : 1 To station : 2 - with a period of 85 seconds.
From station : 1 To station : 5 - with a period of 173 seconds.
From station : 4 To station : 8 - with a period of 183 seconds.
From station : 5 To station : 10 - with a period of 502 seconds.
From station : 5 To station : 1 - with a period of 173 seconds.
From station : 2 To station : 6 - with a period of 80 seconds.
From station : 2 To station : 1 - with a period of 85 seconds.
From station : 6 To station : 9 - with a period of 278 seconds.
From station : 6 To station : 2 - with a period of 80 seconds.
From station : 7 To station : 3 - with a period of 186 seconds.
From station : 3 To station : 7 - with a period of 186 seconds.
From station : 3 To station : 8 - with a period of 103 seconds.
From station : 3 To station : 1 - with a period of 217 seconds.
From station : 8 To station : 10 - with a period of 167 seconds.
From station : 8 To station : 3 - with a period of 91 seconds.
From station : 8 To station : 4 - with a period of 183 seconds.
From station : 10 To station : 8 - with a period of 167 seconds.
From station : 10 To station : 9 - with a period of 84 seconds.
From station : 10 To station : 5 - with a period of 493 seconds.
From station : 9 To station : 6 - with a period of 250 seconds.
From station : 9 To station : 10 - with a period of 84 seconds.
```

Fig. 3 : 'read\_connections' display

The connections between the stations are well read and the time between each is clearly displayed.

## **Part 3 :**

### **Question 8 :**

At the compilation for the first test of 'generic\_mapper.hpp', no error was encountered.

### **Questions 9 and 10 :**

Compute\_travel' and 'Compute\_and\_display\_travel' have been implemented following the Dijkstra algorithm. The algorithm returns the shortest path between two stations of the Parisian underground network, with the necessary stops and the estimated total time. The algorithm has been implemented with a very simple approach which consists in closing the already visited nodes and considering only the shortest time between two stations.

```
(base) emir@MacBook-Air-de-Emir Resources-20230329 % ./main.exe
Shortest way between 8 and 2 :
From station : 8 to station : 3
From station : 3 to station : 1
From station : 1 to station : 2
Total accumulated time : 6 minutes, 33 seconds
```

Fig. 4 : 'compute\_travel' test with station ids

Thus the displays are made according to the 'id' of the given stations. This first step uses the files 'stations.csv' and 'connections.csv'.

To have a clear display, I have turned the time in minutes.

### Question 12 :

For this last 'Bonus' part of the project, the goal was to have a real look at what the application can provide us. So I adapted the format of the two previous functions by overloading them in order to work with the real station names. To do this, I had to normalize the station names by putting them in lower case to avoid conflicts in the search. And so by adding two extra functions to the project, one to normalize the name and one to search for the id of the station by the normalized name, I used the previous 'Compute\_travel' and 'Compute\_and\_display\_travel' functions to optimize the implementation. Moreover, this step uses the files 's.csv' and 'c.csv'.

Here's an example of test between Barbès and Liberté with the shortest path :

```
(base) emir@MacBook-Air-de-Emir Resources-20230329 % ./main.exe
compute_and_display_travel :
Direct path found :
Departure from station -> Barbès-Rochechouart (line 4, Direction => (PORTE DE CLIGNANCOURT <-> MAIRIE DE MONTRouGE) - Retour)
Stopover at -> Gare de l'Est (Verdun) | From line : 4 to line : 5
Take line : 5, Direction => (BOBIGNY - PABLO PICASSO <-> PLACE D'ITALIE) - Aller)
Stopover at -> République | From line : 5 to line : 8
Take line : 8, Direction => (BALARD <-> POINTE DU LAC) - Aller)
Destination reached >> Liberté (line 8)
Total accumulated time : 36 minutes, 49 seconds
```

Fig. 5 : 'compute\_travel' test with station names