

# Oyster Card Problem

1- Running the test case and show balance.

All you need to do is move to the [src](#) directory and run the main.py file. The file will start running showing the steps of the test case, see the following figure. There a delay 1 second to make the output readable. The system will show how the fare calculated and if there are many possible fares, it will favour the customer with the lowest one.

```
(Thaki_project) E:\AlefTask>python src/main.py
Charge the Oyster Card with 30 £.
Card Balance:  30.0 £
-----
Trip 1: From Tube Holborn to Earl's Court
The Customer swipe in at the Tube Holborn Station:
Card balance after charge the max cost:  26.8 £
Possible Fares are:  [2.5, 3.0]
The system will favour the customer with fare:  2.5
The Customer swipe out at the Earl's Court Station:
Current Balance after swipe out:  27.5 £
-----
Trip 2: 328 bus from Earl's Court to Chelsea
The customer swipe in at the 328 bus station.
The Customer will charge only 1.8 £ for all bus journeys.
Card Balance:  25.7 £
-----
Trip 3: From Tube Earl's court to Hammersmith
```

2- Testing the model and coverage

Used the most know unittest and coverage packages to test the system and check if every single method is covered. Go to the [htmlcov](#) folder and open [index.html](#) file to check.

3- Dependencies

The system is created using *python3.8*, *unittest*, *coverage*

Solution by Ahmed Akl