Instruction to candidates:

Your program code and output for each of Task 1 to N should be saved in a single `.ipynb` file using Jupyter Notebook. For example, your program code and output for Task 1 should be saved as:

```
Task1_<your name>_<centre number>_<index number>.ipynb
```

Make sure that each of your `.ipynb` files shows the required output in Jupyter Notebook.

1 Name your Jupyter Notebook as

```
Task4_<your name>_<centre number>_<index number>.ipynb
```

The manager of a private carpark wants to process the daily data of people using the carpark. The carpark is open from 8am to 11pm and the carpark charge is as follows:

- Before 5pm: $1.50 per hour or part thereof
- After 5pm, $3.00 per entry regardless of the duration

However, if a car enters before 5pm and leaves after 5pm, the charge involves both rules. For example, if the car stays in the carpark from 2.45pm to 6.30pm, it is broken down to 2 hours 15 minutes before 5pm and 1 hour 30 minutes after 5pm. Hence the final charges incurred will be $1.50 × 3 + $3.00 = $7.50.

Each day the carpark electronic system generates a file `CARPARK.txt`. Each record in the file has the following format:

```
<CARPLATE NUMBER>,<START TIME>,<END TIME>
```

For example, `SLX2315A,0940,1415` means that a car with car plate number SLX2315A entered the carpark at 9.40am and left at 2.15pm.

You are **not** allowed to use any built-in functions for time processing.

For each of the sub-tasks, add a comment statement at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

```
In [1] :   # Task 4.1
           Program code

           Output:
```

Task 4.1

Write program code for a function `Price` using the following specification:

```
FUNCTION Price (start: STRING, end: STRING) : FLOAT
```

The function has two string parameters `start` and `end`, which refers to the start time and end time for which the car was parked. The function returns the carpark charges incurred as a float.

[5]

Task 4.2

Write program code to perform the following actions for the manager:

- read data from `CARPARK.txt`
- write all the car plate numbers and corresponding carpark charges to another file `CHARGE.txt`, where each line contains a car plate number and the corresponding charges in the format `<CARPLATE NUMBER>,<CARPARK CHARGE>`
- output the total charges for the day.

[10]

Save your Jupyter Notebook for Task 4.