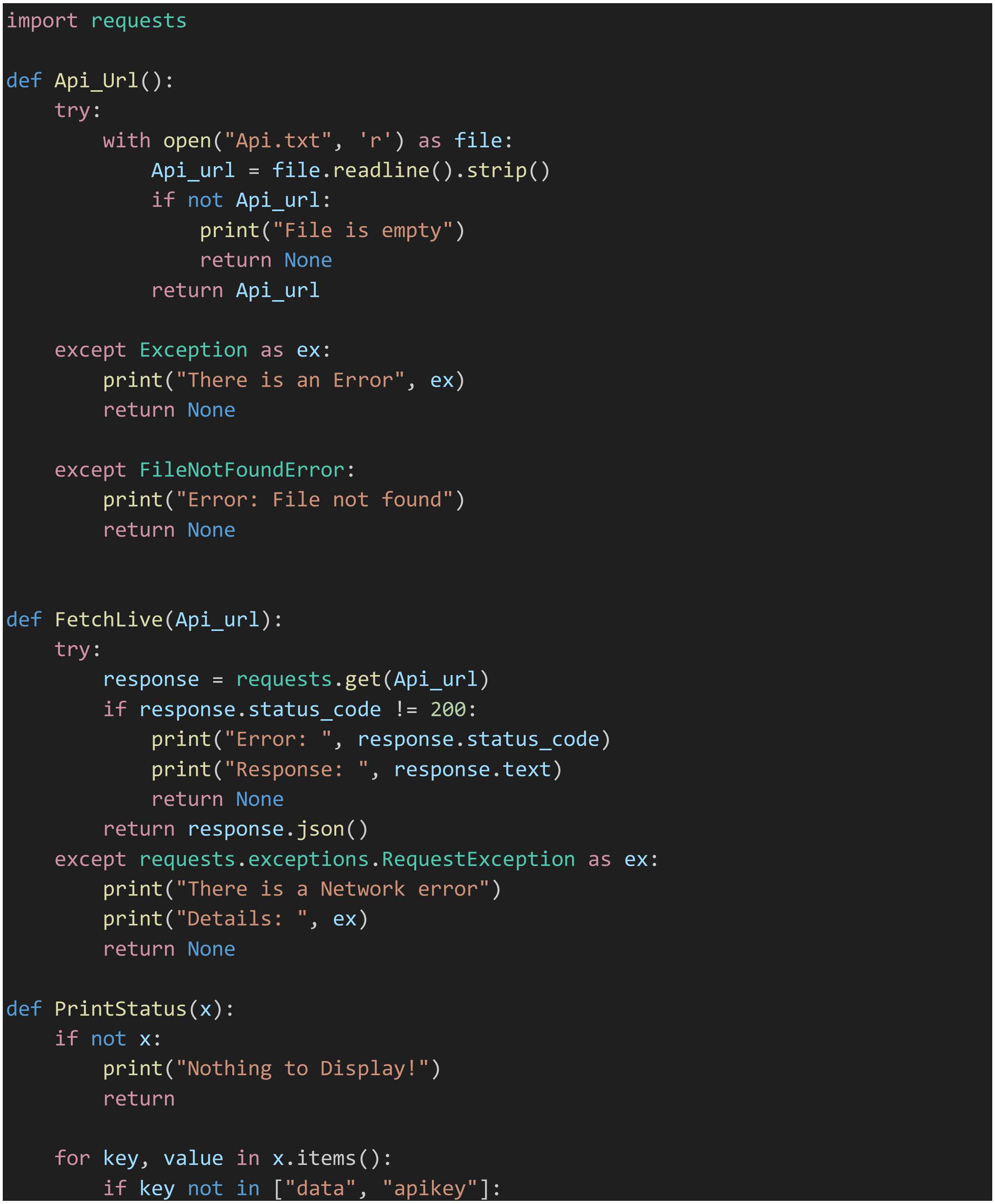
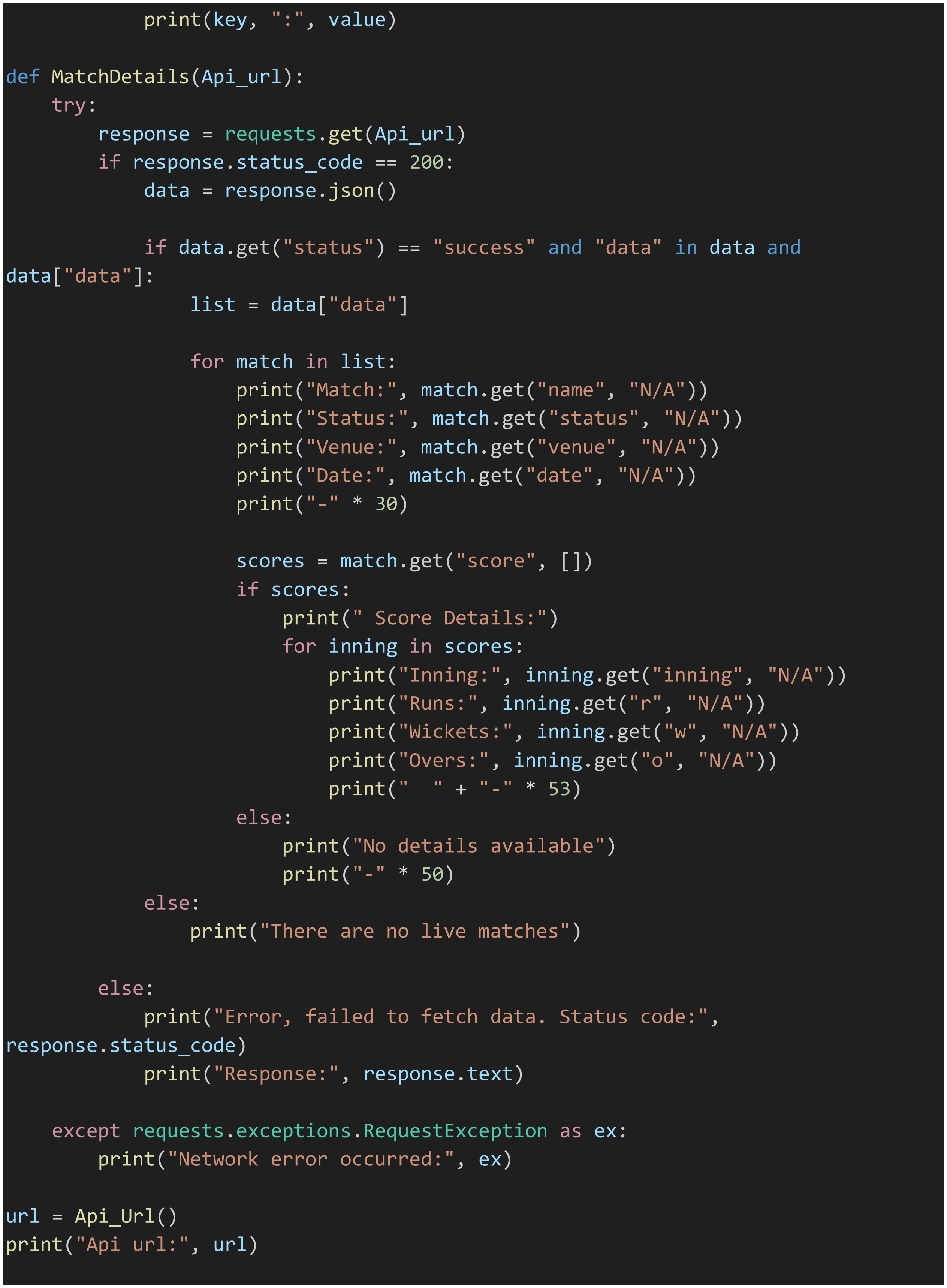
PAI ASSIGNMENT 2

24K-0008

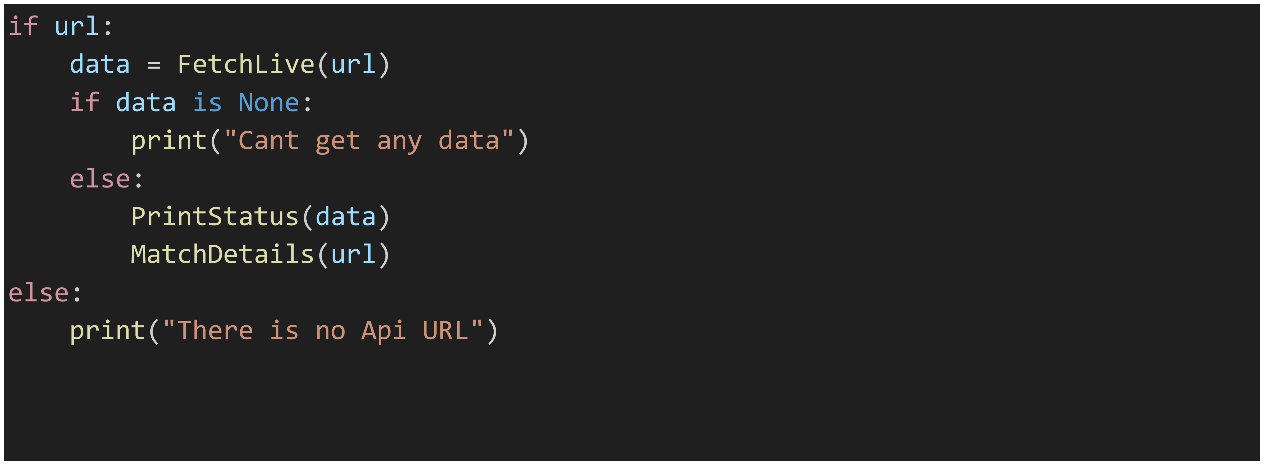
Q1.



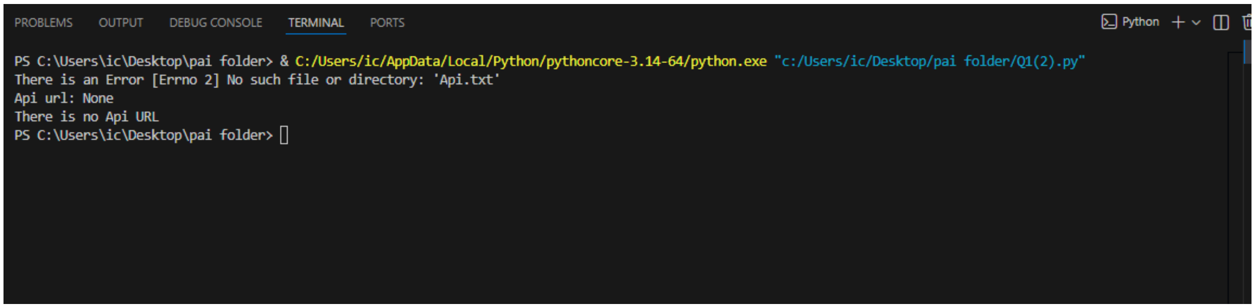
```python
import requests

def Api_Url():
    try:
        with open("Api.txt", 'r') as file:
            Api_url = file.readline().strip()
            if not Api_url:
                print("File is empty")
                return None
            return Api_url

    except Exception as ex:
        print("There is an Error", ex)
        return None

    except FileNotFoundError:
        print("Error: File not found")
        return None


def FetchLive(Api_url):
    try:
        response = requests.get(Api_url)
        if response.status_code != 200:
            print("Error: ", response.status_code)
            print("Response: ", response.text)
            return None
        return response.json()
    except requests.exceptions.RequestException as ex:
        print("There is a Network error")
        print("Details: ", ex)
        return None

def PrintStatus(x):
    if not x:
        print("Nothing to Display!")
        return

    for key, value in x.items():
        if key not in ["data", "apikey"]:
```



```
            print(key, ":", value)

def MatchDetails(Api_url):
    try:
        response = requests.get(Api_url)
        if response.status_code == 200:
            data = response.json()

            if data.get("status") == "success" and "data" in data and 
data["data"]:
                list = data["data"]

                for match in list:
                    print("Match:", match.get("name", "N/A"))
                    print("Status:", match.get("status", "N/A"))
                    print("Venue:", match.get("venue", "N/A"))
                    print("Date:", match.get("date", "N/A"))
                    print("-" * 30)

                    scores = match.get("score", [])
                    if scores:
                        print(" Score Details:")
                        for inning in scores:
                            print("Inning:", inning.get("inning", "N/A"))
                            print("Runs:", inning.get("r", "N/A"))
                            print("Wickets:", inning.get("w", "N/A"))
                            print("Overs:", inning.get("o", "N/A"))
                            print("  " + "-" * 53)
                    else:
                        print("No details available")
                        print("-" * 50)
            else:
                print("There are no live matches")

        else:
            print("Error, failed to fetch data. Status code:", 
response.status_code)
            print("Response:", response.text)

    except requests.exceptions.RequestException as ex:
        print("Network error occurred:", ex)

url = Api_Url()
print("Api url:", url)
```

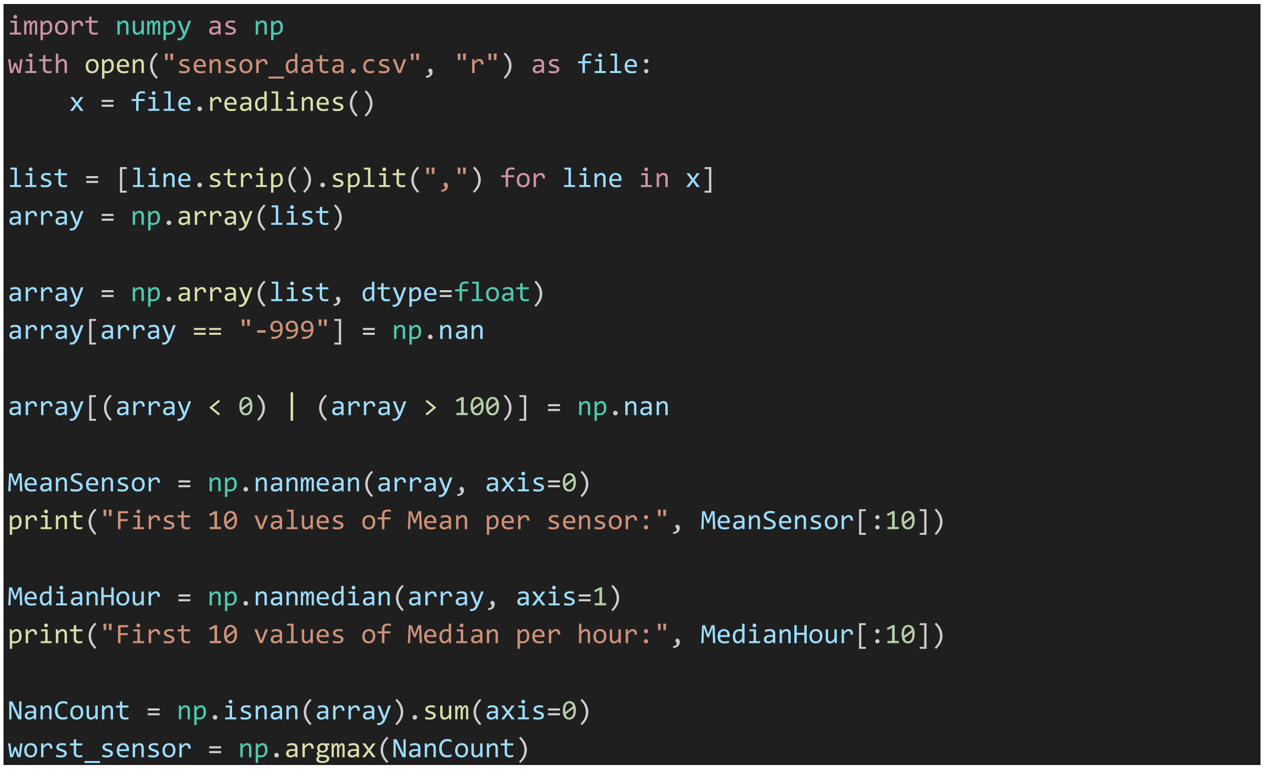


```
if url:
    data = FetchLive(url)
    if data is None:
        print("Cant get any data")
    else:
        PrintStatus(data)
        MatchDetails(url)
else:
    print("There is no Api URL")
```



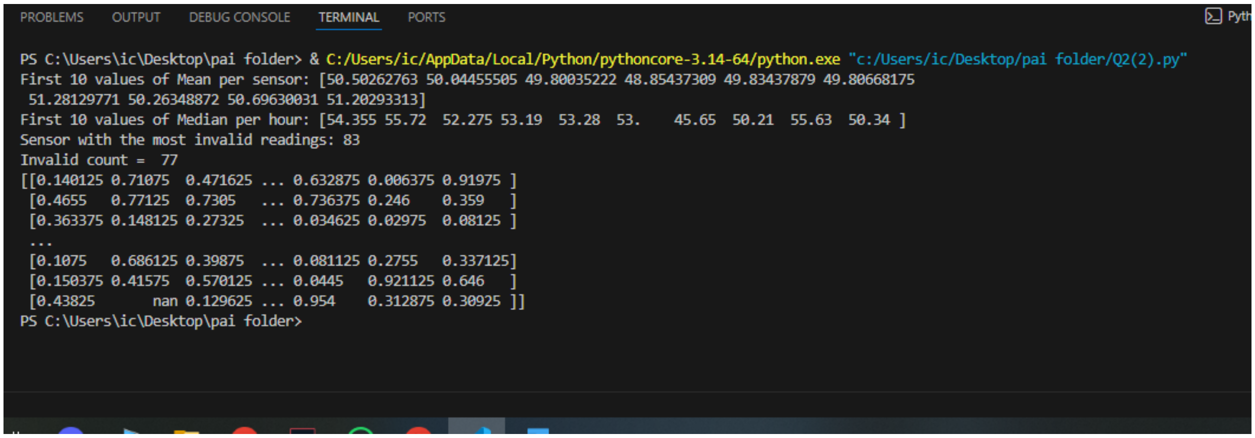
```
PS C:\Users\ic\Desktop\pai folder> & C:/Users/ic/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/ic/Desktop/pai folder/Q1(2).py"
There is an Error [Errno 2] No such file or directory: 'Api.txt'
Api url: None
There is no Api URL
PS C:\Users\ic\Desktop\pai folder> 
```

Q2.



```
import numpy as np
with open("sensor_data.csv", "r") as file:
    x = file.readlines()

list = [line.strip().split(",") for line in x]
array = np.array(list)

array = np.array(list, dtype=float)
array[array == "-999"] = np.nan

array[(array < 0) | (array > 100)] = np.nan

MeanSensor = np.nanmean(array, axis=0)
print("First 10 values of Mean per sensor:", MeanSensor[:10])

MedianHour = np.nanmedian(array, axis=1)
print("First 10 values of Median per hour:", MedianHour[:10])

NanCount = np.isnan(array).sum(axis=0)
worst_sensor = np.argmax(NanCount)
```

```
print("Sensor with the most invalid readings:", worst_sensor)
print("Invalid count = ", NanCount[worst_sensor])

xmin = np.nanmin(array)
xmax = np.nanmax(array)
xnorm = (array - xmin) / (xmax - xmin)
print(xnorm)

np.savetxt("sensor_data_normalized.csv", xnorm)
```

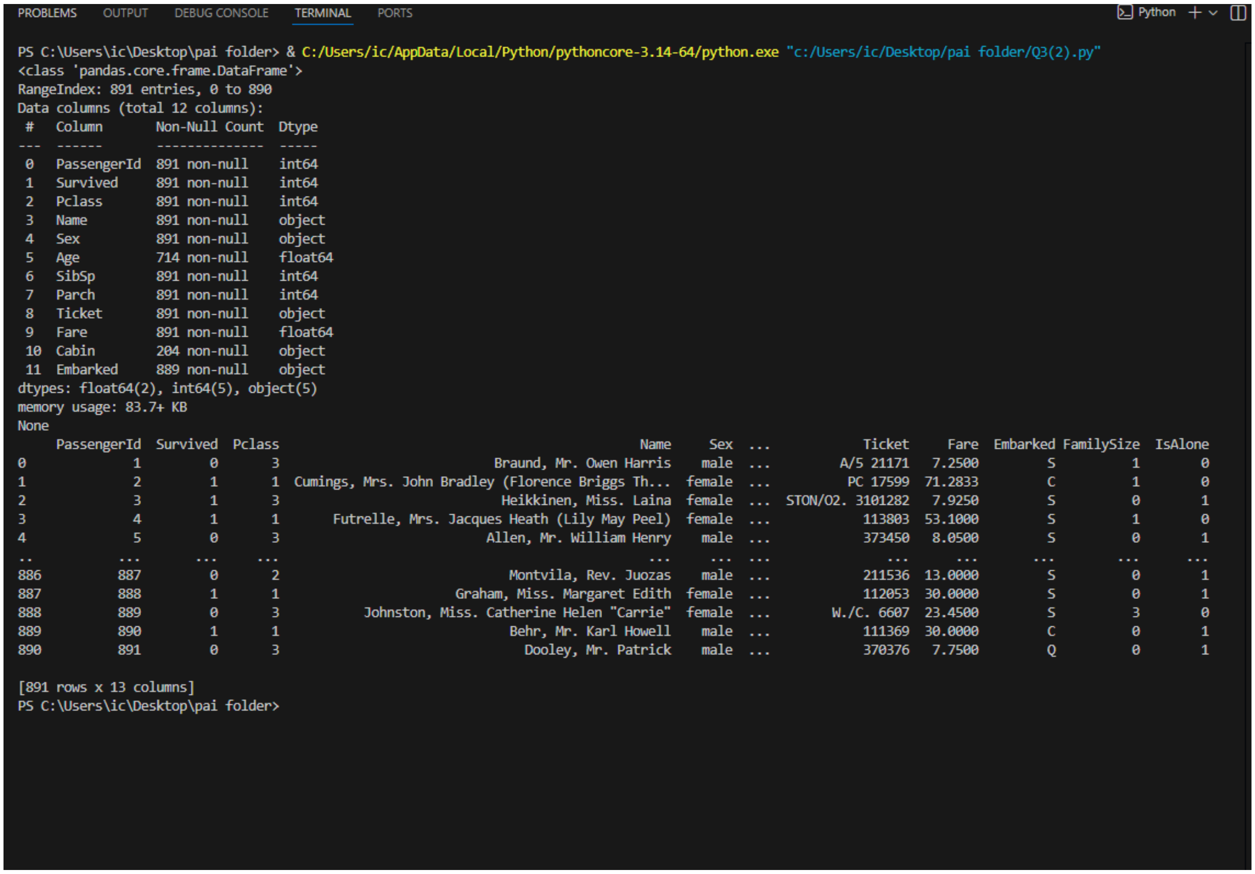


```
PS C:\Users\ic\Desktop\pai folder> & C:/Users/ic/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/ic/Desktop/pai folder/Q2(2).py"
First 10 values of Mean per sensor: [50.50262763 50.04455505 49.80035222 48.85437309 49.83437879 49.80668175
 51.28129771 50.26348872 50.69630031 51.20293313]
First 10 values of Median per hour: [54.355 55.72  52.275 53.19  53.28  53.    45.65  50.21  55.63  50.34 ]
Sensor with the most invalid readings: 83
Invalid count =  77
[[0.140125 0.71075  0.471625 ... 0.632875 0.006375 0.91975 ]
 [0.4655   0.77125  0.7305   ... 0.736375 0.246    0.359   ]
 [0.363375 0.148125 0.27325  ... 0.034625 0.02975  0.08125 ]
 ...
 [0.1075   0.686125 0.39875  ... 0.081125 0.2755   0.337125]
 [0.150375 0.41575  0.570125 ... 0.0445   0.921125 0.646   ]
 [0.43825       nan 0.129625 ... 0.954    0.312875 0.30925 ]]
PS C:\Users\ic\Desktop\pai folder>
```

Q3.

```
import pandas as pd
df=pd.read_csv("Titanic-Dataset.csv")


report=df.info()
print(report)

df["Age"]=df["Age"].fillna(df.groupby(["Pclass","Sex"])["Age"].transform("median"
))
df["Embarked"]=df["Embarked"].fillna(df["Embarked"].mode()[0])
df=df.drop("Cabin",axis=1)

df.insert(11,"FamilySize",df["SibSp"] + df["Parch"])

df["IsAlone"] = 0
df.loc[df['FamilySize'] == 0, 'IsAlone'] = 1


df["Age"]=df["Age"].astype(int)
```

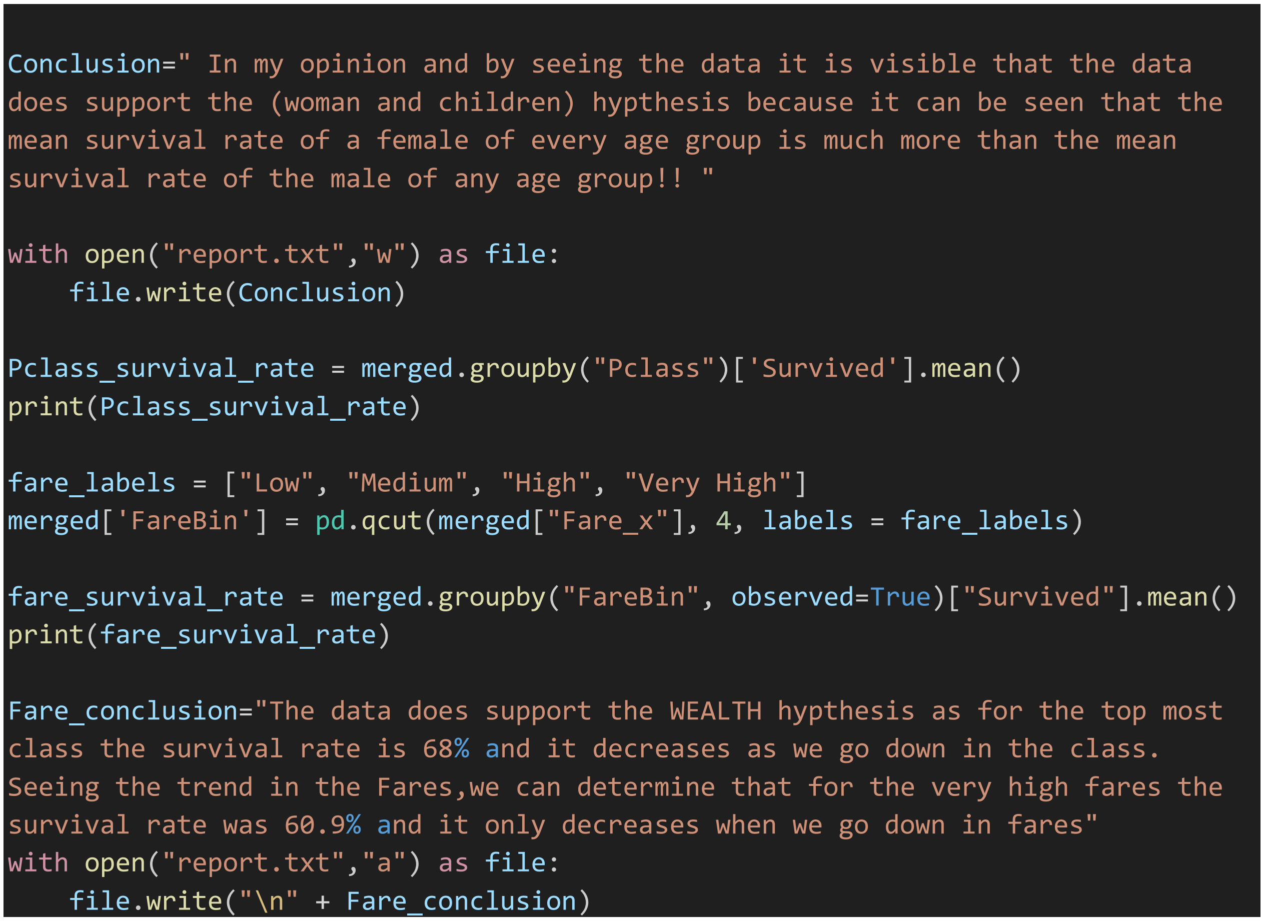
```python
print(df)

df.to_csv("titanic_cleaned.csv",index=False)
```



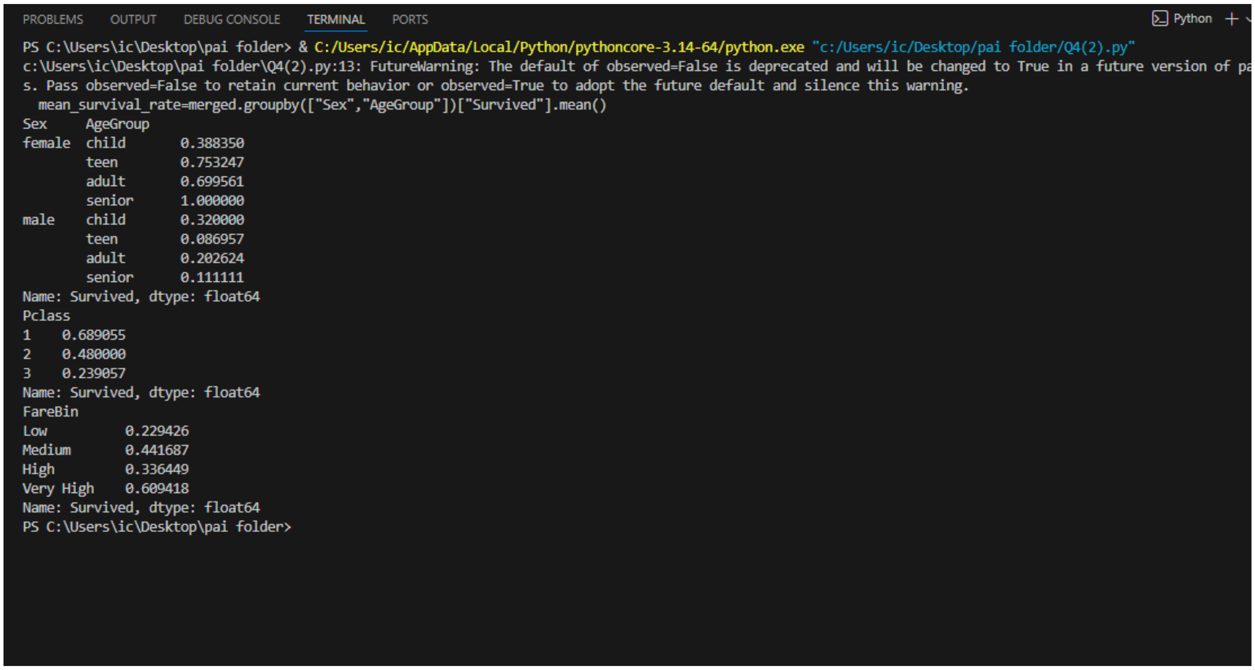
```
PS C:\Users\ic\Desktop\pai folder> & C:/Users/ic/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/ic/Desktop/pai folder/Q3(2).py"
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
None
     PassengerId  Survived  Pclass                                                Name     Sex  ...            Ticket     Fare  Embarked FamilySize  IsAlone
0              1         0       3                             Braund, Mr. Owen Harris    male  ...         A/5 21171   7.2500        S          1        0
1              2         1       1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  ...          PC 17599  71.2833        C          1        0
2              3         1       3                              Heikkinen, Miss. Laina  female  ...  STON/O2. 3101282   7.9250        S          0        1
3              4         1       1        Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  ...            113803  53.1000        S          1        0
4              5         0       3                            Allen, Mr. William Henry    male  ...            373450   8.0500        S          0        1
..           ...       ...     ...                                                 ...     ...  ...               ...      ...       ...        ...      ...
886          887         0       2                               Montvila, Rev. Juozas    male  ...            211536  13.0000        S          0        1
887          888         1       1                        Graham, Miss. Margaret Edith  female  ...            112053  30.0000        S          0        1
888          889         0       3            Johnston, Miss. Catherine Helen "Carrie"  female  ...        W./C. 6607  23.4500        S          3        0
889          890         1       1                               Behr, Mr. Karl Howell    male  ...            111369  30.0000        C          0        1
890          891         0       3                                 Dooley, Mr. Patrick    male  ...            370376   7.7500        Q          0        1

[891 rows x 13 columns]
PS C:\Users\ic\Desktop\pai folder>
```

Q4.

```python
import pandas as pd
import numpy as np

df=pd.read_csv("titanic_cleaned.csv")
dd=pd.read_csv("ticket_fares.csv")

merged=pd.merge(df,dd,on="Ticket",how="inner")

bins=[0,12,19,59,120]
labels=["child","teen","adult","senior"]
merged["AgeGroup"]=pd.cut(merged["Age"],bins=bins,labels=labels,right=True)

mean_survival_rate=merged.groupby(["Sex","AgeGroup"])["Survived"].mean()
print(mean_survival_rate)
```



```
Conclusion=" In my opinion and by seeing the data it is visible that the data
does support the (woman and children) hypthesis because it can be seen that the
mean survival rate of a female of every age group is much more than the mean
survival rate of the male of any age group!! "

with open("report.txt","w") as file:
    file.write(Conclusion)

Pclass_survival_rate = merged.groupby("Pclass")['Survived'].mean()
print(Pclass_survival_rate)

fare_labels = ["Low", "Medium", "High", "Very High"]
merged['FareBin'] = pd.qcut(merged["Fare_x"], 4, labels = fare_labels)

fare_survival_rate = merged.groupby("FareBin", observed=True)["Survived"].mean()
print(fare_survival_rate)

Fare_conclusion="The data does support the WEALTH hypthesis as for the top most
class the survival rate is 68% and it decreases as we go down in the class.
Seeing the trend in the Fares,we can determine that for the very high fares the
survival rate was 60.9% and it only decreases when we go down in fares"
with open("report.txt","a") as file:
    file.write("\n" + Fare_conclusion)
```



```
PS C:\Users\ic\Desktop\pai folder> & C:/Users/ic/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/ic/Desktop/pai folder/Q4(2).py"
c:\Users\ic\Desktop\pai folder\Q4(2).py:13: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pa
s. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  mean_survival_rate=merged.groupby(["Sex","AgeGroup"])["Survived"].mean()
Sex     AgeGroup
female  child       0.388350
        teen        0.753247
        adult       0.699561
        senior      1.000000
male    child       0.320000
        teen        0.086957
        adult       0.202624
        senior      0.111111
Name: Survived, dtype: float64
Pclass
1    0.689055
2    0.480000
3    0.239057
Name: Survived, dtype: float64
FareBin
Low          0.229426
Medium       0.441687
High         0.336449
Very High    0.609418
Name: Survived, dtype: float64
PS C:\Users\ic\Desktop\pai folder>
```