Import pandas and read the data into a DataFrame. Print the first 5 rows of the DataFrame

In [1]:
```python
import pandas as pd
import numpy as np
```

In [2]:
```python
df=pd.read_csv('padas_question.csv')
```

In [6]:
```python
df.head(5)
```

Out[6]:

	Name	Gender	Age	Height (cm)	Weight (kg)	City
0	Alice	Female	25	165	58	New York
1	Bob	Male	30	180	80	Los Angeles
2	Charlie	Male	35	175	70	Chicago
3	Diana	Female	28	170	65	Miami
4	Eva	Female	27	175	60	New York

Print the number of rows and columns in the DataFrame

In [10]:
```python
row=len(df.index)
column=len(df.columns)
```

In [13]:
```python
print('Total Row is: ',row)
print('Total column is: ',column)
```

```
Total Row is:  10
Total column is:  6
```

Print the null value count

In [14]:
```python
null_value=df.isnull()
```

In [15]:
```python
null_value
```

Out[15]:

	Name	Gender	Age	Height (cm)	Weight (kg)	City
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
5	False	False	False	False	False	False
6	False	False	False	False	False	False
7	False	False	False	False	False	False
8	False	False	False	False	False	False
9	False	False	False	False	False	False

4.Print the average age, height, and weight of the people in the DataFrame

In [22]:
```python
avg=df['Age'].mean()
height = df['Height (cm)'].mean()
wt = df['Weight (kg)'].mean()
```

In [23]:
```python
print('Average age is ',avg)
print('Average height is ',height)
print('Average weight is ',wt)
```

```
Average age is  31.6
Average height is  175.0
Average weight is  68.9
```

5. Print the name and city of the person who is the tallest in the DataFrame.

In [30]:
```python
taller=df.sort_values('Height (cm)',ascending=False)
```

In [35]:
```python
print ('name and city of the person who is the tallest \n ',taller)
```

```
name and city of the person who is the tallest
      Name  Gender  Age  Height (cm)  Weight (kg)         City
5    Frank    Male   40          190           85      Houston
1      Bob    Male   30          180           80  Los Angeles
9     Jack    Male   28          180           78     New York
2  Charlie    Male   35          175           70      Chicago
4      Eva  Female   27          175           60     New York
7    Henry    Male   45          175           75  Los Angeles
8      Ivy  Female   26          172           63        Miami
3    Diana  Female   28          170           65        Miami
6    Grace  Female   32          168           55      Chicago
0    Alice  Female   25          165           58     New York
```

6. Create a new column in the DataFrame called "BMI" that calculates the body mass index of each person. BMI is calculated as weight (kg) divided by height (m) squared. Print the DataFrame with the new column.

```
In [44]: BMI=df['Weight (kg)']/(df['Height (cm)']/100)**2
```

```
In [46]: df['BMI'] = BMI
```

```
In [50]: print (df)
```

```
        Name  Gender  Age  Height (cm)  Weight (kg)        City        BMI
0      Alice  Female   25          165           58    New York  21.303949
1        Bob    Male   30          180           80  Los Angeles  24.691358
2    Charlie    Male   35          175           70      Chicago  22.857143
3      Diana  Female   28          170           65        Miami  22.491349
4        Eva  Female   27          175           60    New York  19.591837
5      Frank    Male   40          190           85      Houston  23.545706
6      Grace  Female   32          168           55      Chicago  19.486961
7      Henry    Male   45          175           75  Los Angeles  24.489796
8        Ivy  Female   26          172           63        Miami  21.295295
9       Jack    Male   28          180           78    New York  24.074074
```

7. Group the data by gender and print the average age, height, and weight of males and females separately.

```
In [65]: gup = df.groupby(['Gender']).mean()
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_3596\2301725546.py:1: FutureWarning: Th
e default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a futur
e version, numeric_only will default to False. Either specify numeric_only or sele
ct only columns which should be valid for the function.
  gup = df.groupby(['Gender']).mean()
```

```
In [64]: print (gup)
```

```
         Age  Height (cm)  Weight (kg)        BMI
Gender
Female  27.6        170.0         60.2  20.833878
Male    35.6        180.0         77.6  23.931615
```

8. Sort the data by age in ascending order and print the first 3 rows of the sorted DataFrame.

```
In [66]: ageaz=df.sort_values('Age')
```

```
In [69]: print(ageaz.head(3))
```

```
    Name  Gender  Age  Height (cm)  Weight (kg)      City        BMI
0  Alice  Female   25          165           58  New York  21.303949
8    Ivy  Female   26          172           63     Miami  21.295295
4    Eva  Female   27          175           60  New York  19.591837
```

9. Filter the data to only include people who live in New York and print the resulting DataFrame.

```
In [83]: newyork = df.loc[df['City'] == 'New York']
```

```
In [85]: print (newyork)
```

```
     Name  Gender  Age  Height (cm)  Weight (kg)      City       BMI
0   Alice  Female   25          165           58  New York  21.303949
4     Eva  Female   27          175           60  New York  19.591837
9    Jack    Male   28          180           78  New York  24.074074
```

In [ ]: