A) Write a Python program for Handling Missing Value. Replace missing value of salary, age column with mean of that column.(Use Data.csv file).

In [3]:

```
# import libraries
import pandas as pd
```

In [4]:

```
# read  Iris dataset
df=pd.read_csv("D://Datasets/Datascale.csv")
df
```

Out[4]:

| | Country | Age | Salary | Purchased |
|---|---|---|---|---|
| **0** | France | 44.0 | 72000.0 | No |
| **1** | Spain | 27.0 | 48000.0 | Yes |
| **2** | Germany | 30.0 | 54000.0 | No |
| **3** | Spain | 38.0 | 61000.0 | No |
| **4** | Germany | 40.0 | NaN | Yes |
| **5** | France | 35.0 | 58000.0 | Yes |
| **6** | Spain | NaN | 52000.0 | No |
| **7** | France | 48.0 | 79000.0 | Yes |
| **8** | Germany | 50.0 | 83000.0 | No |
| **9** | France | 37.0 | 67000.0 | Yes |

In [5]:

```
#finding missing values
df.isnull().sum()
```

Out[5]:

```
Country      0
Age          1
Salary       1
Purchased    0
dtype: int64
```

In [6]:

```
# Replacing null values of salary by its mean values
smean=round(df['Salary'].mean(),2)
print("mean of Salary=",end="")
print(smean)
print("Data after filling null value of Salary column")
df['Salary']=df['Salary'].fillna(smean)
df
```

```
mean of Salary=63777.78
Data after filling null value of Salary column
```

Out[6]:

| | Country | Age | Salary | Purchased |
|---|---|---|---|---|
| **0** | France | 44.0 | 72000.00 | No |
| **1** | Spain | 27.0 | 48000.00 | Yes |
| **2** | Germany | 30.0 | 54000.00 | No |
| **3** | Spain | 38.0 | 61000.00 | No |
| **4** | Germany | 40.0 | 63777.78 | Yes |
| **5** | France | 35.0 | 58000.00 | Yes |
| **6** | Spain | NaN | 52000.00 | No |
| **7** | France | 48.0 | 79000.00 | Yes |
| **8** | Germany | 50.0 | 83000.00 | No |
| **9** | France | 37.0 | 67000.00 | Yes |

In [7]:

```
# Replacing null values of salary by its mean values
agemean=round(df['Age'].mean(),2)
print("mean of column Age=",end="")
print(agemean)

print("Data after filling null value of Age column")
df['Age']=df['Age'].fillna(agemean)
df
```

```
mean of column Age=38.78
Data after filling null value of Age column
```

Out[7]:

| | Country | Age | Salary | Purchased |
|---|---|---|---|---|
| **0** | France | 44.00 | 72000.00 | No |
| **1** | Spain | 27.00 | 48000.00 | Yes |
| **2** | Germany | 30.00 | 54000.00 | No |
| **3** | Spain | 38.00 | 61000.00 | No |
| **4** | Germany | 40.00 | 63777.78 | Yes |
| **5** | France | 35.00 | 58000.00 | Yes |
| **6** | Spain | 38.78 | 52000.00 | No |
| **7** | France | 48.00 | 79000.00 | Yes |
| **8** | Germany | 50.00 | 83000.00 | No |
| **9** | France | 37.00 | 67000.00 | Yes |

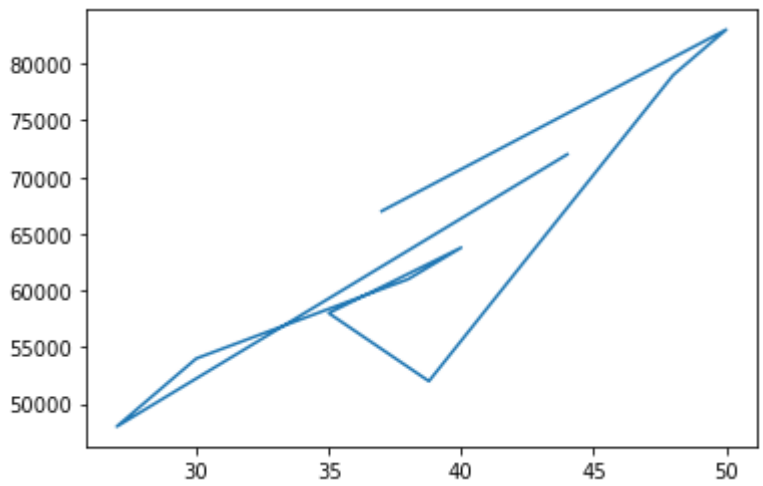
B) Write a Python program to generate a line plot of Age Vs salary

In [10]:

```
import matplotlib.pyplot as plt
plt.plot(df['Age'],df['Salary'])
```

Out[10]:

```
[<matplotlib.lines.Line2D at 0xb1d7aaa070>]
```




C) Download the heights and weights dataset and load the dataset froma given csv file into a dataframe. Print the first, last 10 rows and random 20 rows also display shape of the dataset

In [16]:

```
df1=pd.read_csv("D://Datasets/HeightWeight.csv")
df1
```

Out[16]:

| | Index | Height(Inches) | Weight(Pounds) |
|---|---|---|---|
| **0** | 1 | 65.78331 | 112.9925 |
| **1** | 2 | 71.51521 | 136.4873 |
| **2** | 3 | 69.39874 | 153.0269 |
| **3** | 4 | 68.21660 | 142.3354 |
| **4** | 5 | 67.78781 | 144.2971 |
| **...** | ... | ... | ... |
| **24995** | 24996 | 69.50215 | 118.0312 |
| **24996** | 24997 | 64.54826 | 120.1932 |
| **24997** | 24998 | 64.69855 | 118.2655 |
| **24998** | 24999 | 67.52918 | 132.2682 |
| **24999** | 25000 | 68.87761 | 124.8742 |

25000 rows × 3 columns

In [17]:

```
#first 10 records
df1.head(10)
```

Out[17]:

| | Index | Height(Inches) | Weight(Pounds) |
|---|---|---|---|
| **0** | 1 | 65.78331 | 112.9925 |
| **1** | 2 | 71.51521 | 136.4873 |
| **2** | 3 | 69.39874 | 153.0269 |
| **3** | 4 | 68.21660 | 142.3354 |
| **4** | 5 | 67.78781 | 144.2971 |
| **5** | 6 | 68.69784 | 123.3024 |
| **6** | 7 | 69.80204 | 141.4947 |
| **7** | 8 | 70.01472 | 136.4623 |
| **8** | 9 | 67.90265 | 112.3723 |
| **9** | 10 | 66.78236 | 120.6672 |

In [18]:

```
#Last 10 records
df1.tail(10)
```

Out[18]:

| | Index | Height(Inches) | Weight(Pounds) |
|---|---|---|---|
| **24990** | 24991 | 69.97767 | 125.3672 |
| **24991** | 24992 | 71.91656 | 128.2840 |
| **24992** | 24993 | 70.96218 | 146.1936 |
| **24993** | 24994 | 66.19462 | 118.7974 |
| **24994** | 24995 | 67.21126 | 127.6603 |
| **24995** | 24996 | 69.50215 | 118.0312 |
| **24996** | 24997 | 64.54826 | 120.1932 |
| **24997** | 24998 | 64.69855 | 118.2655 |
| **24998** | 24999 | 67.52918 | 132.2682 |
| **24999** | 25000 | 68.87761 | 124.8742 |

In [19]:

```
# Random 20 records
df1.sample(20)
```

Out[19]:

| | Index | Height(Inches) | Weight(Pounds) |
|---|---|---|---|
| **5548** | 5549 | 67.53327 | 120.1058 |
| **6493** | 6494 | 70.09165 | 152.4262 |
| **16466** | 16467 | 67.84324 | 144.1374 |
| **3525** | 3526 | 67.15633 | 112.9380 |
| **910** | 911 | 66.46981 | 141.4417 |
| **22854** | 22855 | 73.28207 | 137.8167 |
| **19535** | 19536 | 67.93550 | 133.4828 |
| **2449** | 2450 | 64.65154 | 124.8094 |
| **8982** | 8983 | 67.74427 | 124.1332 |
| **24897** | 24898 | 66.99191 | 142.8088 |
| **1511** | 1512 | 65.40895 | 110.5741 |
| **11814** | 11815 | 68.32665 | 139.0646 |
| **6279** | 6280 | 68.39180 | 123.9172 |
| **8267** | 8268 | 68.61117 | 129.6617 |
| **1858** | 1859 | 72.48112 | 146.3046 |
| **14567** | 14568 | 66.53695 | 122.4950 |
| **12115** | 12116 | 67.45327 | 124.3866 |
| **10462** | 10463 | 66.11571 | 136.9935 |
| **9544** | 9545 | 69.19418 | 122.8981 |
| **18903** | 18904 | 72.84287 | 142.1741 |

In [20]:

```
#shape of dataset
df1.shape
```

Out[20]:

```
(25000, 3)
```

In [ ]:

```
agemean=df['age'].mean()
df['age']=df.['age'].fillna(agemean)
```