Assignment 3: Assignment based on Calculations of Central Tendency Measures (Mean, Median, Mode)

Without Using Inbuilt Functions

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

In [3]:
```
# Program to calculate Mean, Median and Mode without using inbuilt functions
df = pd.read_csv('student_marks.csv')
```

In [4]:
```
df.head()
```

Out[4]:

	Student	Marks
0	S1	78
1	S2	65
2	S3	80
3	S4	92
4	S5	85

In [5]:
```
df.tail()
```

Out[5]:

	Student	Marks
5	S6	75
6	S7	88
7	S8	90
8	S9	70
9	S10	85

In [12]:
```
# Mean Calculation
def calculate_mean(data):
    return sum(data) / len(data)

mean = calculate_mean(df['Marks'])
print(mean)
```

```
80.8
```

In [14]:
```
# Median Calculation
def calculate_median(data):
    sorted_data = sorted(data)
    n = len(sorted_data)
    mid = n // 2
    if n % 2 == 0:
```

```
        median = (sorted_data[mid-1] + sorted_data[mid]) / 2
    else:
        median = sorted_data[mid]
    return median

median = calculate_median(df['Marks'])
print(median)
```

```
82.5
```

In [18]:
```
# Mode Calculation
def calculate_mode(data):
    freq = {}
    for val in data:
        freq[val] = freq.get(val, 0) + 1
    max_freq = max(freq.values())
    mode = [key for key, value in freq.items() if value == max_freq]
    return mode

mode = calculate_mode(df['Marks'])
print(mode)
```

```
[85]
```

Using Inbuilt Functions

In [19]:
```
import statistics as stats

data = pd.read_csv('student_marks.csv')
```

In [21]:
```
mean_value = stats.mean(data['Marks'])
print(mean)
```

```
80.8
```

In [22]:
```
median_value = stats.median(data['Marks'])
print(median)
```

```
82.5
```

In [23]:
```
mode = stats.mode(data['Marks'])
print(mode)
```

```
85
```

In [ ]: