Name: Muhammad Umer Adeeb

Email: umeradeeb94@gmail.com

Number: 03412114514

Profession: Production Engineer

Organization: Amreli Steels Limited

Q1: Given the list numbers = [10, 20, 30, 40, 50], write a Python code snippet to:

```
#Append the number 60 to the list.
numbers = [10, 20, 30, 40, 50]
numbers.append(60)
print(numbers)

#Remove the number 20 from the list.
numbers.remove(20)
print(numbers)

#Insert 80 at index 2.
numbers.insert(2,80)
print(numbers)

#Print the sum of all the elements in the list.
sum_numbers = sum(numbers)
print('Sum of numbers in list:', sum_numbers)
```

```
[10, 20, 30, 40, 50, 60]
[10, 30, 40, 50, 60]
[10, 30, 80, 40, 50, 60]
Sum of numbers in list: 270
```

Q2: Consider the tuple coordinates = (12.5, 45.8, 33.1).

```
#Write a Python code snippet to unpack the values of the tuple into
three variables: `x`, `y`, and `z`.

x,y,z = (12.5, 45.8, 33.1)
print(x)
print(y)
print(z)

#Explain why tuples are generally preferred over lists for storing
```

```
fixed sets of data.
#Answer: Tuples are generally preferred over lists for storing fixed sets of data because tuples are immutable which means elements in tuples can't be changed and list are mutable allowing modifications.

12.5
45.8
33.1
```

Q3: Given the dictionary `student_scores = {Faraz: 85, Raza: 90, Ishaq: 78}`:

```
#Write a Python code snippet to add a new student `David` with a score of 92.
student_scores = {'Faraz': 85, 'Raza': 90, 'Ishaq': 78}
student_scores['David'] = 92
print(student_scores)

#Update Raza's score to 82.
student_scores['Raza'] = 82
print(student_scores)

#Write a code snippet to print all student names and their scores in the format: `Name: Score`.
for name, score in student_scores.items():
    print(f'{name}:{score}')

{'Faraz': 85, 'Raza': 90, 'Ishaq': 78, 'David': 92}
{'Faraz': 85, 'Raza': 82, 'Ishaq': 78, 'David': 92}
Faraz:85
Raza:82
Ishaq:78
David:92
```

Q4: Write a Python function called `calculate_mean` that takes a list of numbers as input and

returns the mean (average) of the numbers.

```
def calculate_mean(numbers):
    if not numbers:
        return 0
    total_sum = sum(numbers)
    mean = total_sum / len(numbers)
    return mean

def user():
    n = int(input('How many numbers you want to enter:'))
```

```
    numbers = []
    for i in range(n):
        num = float(input('Enter numbers:'))
        numbers.append(num)
    mean = calculate_mean(numbers)
    print(f'The mean of given numbers: {mean}')
user()
```

```
How many numbers you want to enter: 5
Enter numbers: 2
Enter numbers: 2
Enter numbers: 4
Enter numbers: 6
Enter numbers: 8

The mean of given numbers: 4.4
```

Q5: Write a Python function called `grade_students` that takes a dictionary of student names and

their scores, and returns a new dictionary with student names and their corresponding grades (A B, C, D, F). Use the following grading scale:

- A: 90- 100
- B: 8 0-89
- C: 70-79
- D : 60-69
- F: Below 60

```
def grade(percentage):
    if percentage >= 90:
        return 'A'
    elif percentage >= 80:
        return 'B'
    elif percentage >= 70:
        return 'C'
    elif percentage >= 60:
        return 'D'
    else:
        return 'F'

def result():
    n = int(input('How many students names do you want to enter: '))
    students_grade = {}
```

```
    for i in range(n):
        student_name = input('Enter name of student: ')
        student_percentage = float(input(f'Enter percentage for
{student_name}: '))

        grade_value = grade(student_percentage)

        students_grade[student_name] = grade_value

    for student, grade_value in students_grade.items():
        print(f'{student}: {grade_value}')


result()
```

```
How many students names do you want to enter:  3
Enter name of student:  umer
Enter percentage for umer:  88
Enter name of student:  bilal
Enter percentage for bilal:  99
Enter name of student:  hassan
Enter percentage for hassan:  91

umer: B
bilal: A
hassan: A
```

Q6: Given a CSV file named `data.csv` with columns `Name`, `Age`, and `Salary`, write a Python

code snippet using pandas to:

```
import pandas as pd
import numpy as np

#Load the CSV file into a DataFrame.
df = pd.read_csv('dataset.csv')

#Display the first 5 rows of the DataFrame.
df.head()
```

```
      Name  Age  Salary
0    Alice   28   70000
1      Bob   34   80000
2  Charlie   29   65000
3    David   40  120000
4      Eva   22   50000
```

```
#Display the last 5 rows of the DataFrame.
df.tail()
```

```
     Name  Age  Salary
5   Frank   36   95000
6   Grace   30   72000
7  Hannah   25   60000
8     Ian   45  130000
9    Jack   38  110000
```

```
#Print the mean salary and see if any missing values present visualize
the missing values if there.
print(df.isnull().sum())
print('Mean salary:', df['Salary'].mean())
```

```
Name      0
Age       0
Salary    0
dtype: int64
Mean salary: 85200.0
```

Q7: Calculate and print summary statistics for the `Age` column (mean, median, standard

deviation).

```
print(df['Age'].mean())
print(df['Age'].median())
print(df['Age'].std())
print(df['Age'].describe())
```

```
32.7
32.0
7.165503781622367
count    10.000000
mean     32.700000
std       7.165504
min      22.000000
25%      28.250000
50%      32.000000
75%      37.500000
max      45.000000
Name: Age, dtype: float64
```

Q8: Using the same DataFrame, write a Python code snippet to: Filter and print the records of employees who are older than 30 years.

```
filtered_df = df[df['Age']>30]
print(filtered_df)
```

```
    Name  Age  Salary
1    Bob   34   80000
3  David   40  120000
5  Frank   36   95000
8    Ian   45  130000
9   Jack   38  110000
```

Q9: Using the same DataFrame, write a Python code snippet to: Group the data by `Age` and calculate the mean salary for each age group Print the result.

```
group_age = df.groupby('Age')['Salary'].mean()
print(group_age)
```

```
Age
22     50000.0
25     60000.0
28     70000.0
29     65000.0
30     72000.0
34     80000.0
36     95000.0
38    110000.0
40    120000.0
45    130000.0
Name: Salary, dtype: float64
```

Q10:

```
def load_and_describe_csv('dataset.csv'):
df = pd.read_csv('dataset.csv')
summary_stats = df.describe()
    return df, summary_stats
filename = 'dataset.csv'
df, stats = load_and_describe_csv(dataset.csv)
print(df.head())
print(stats)
```

```
  Cell In[33], line 1
    def load_and_describe_csv('dataset.csv'):
                              ^
SyntaxError: invalid syntax
```

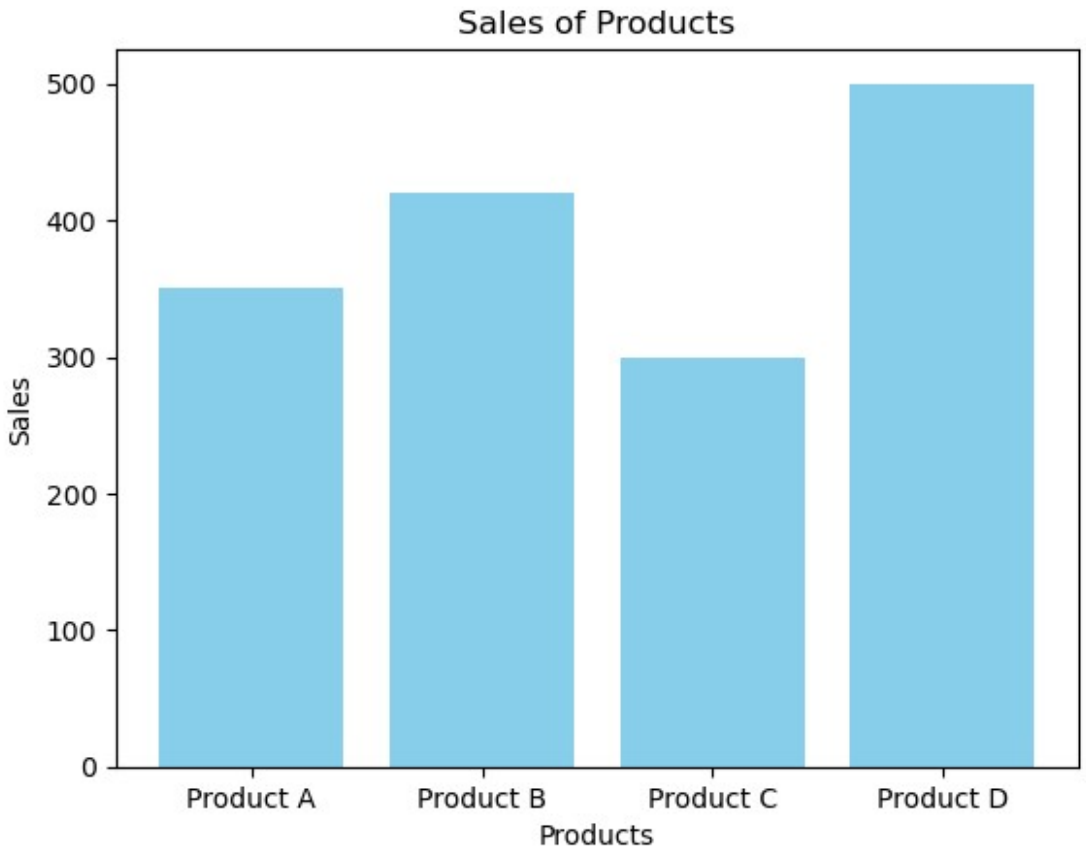
Q11: Bar graph and Pie chart

```
from matplotlib import pyplot as plt

products = ['Product A', 'Product B', 'Product C', 'Product D']
sales = [350, 420, 300, 500]

plt.title('Sales of Products')
plt.xlabel('Products')
plt.ylabel('Sales')
plt.bar(products, sales, color='skyblue')

plt.show()
```




```
plt.pie(sales, labels=products, autopct='%1.1f%%', colors=['gold',
'lightgreen', 'lightcoral', 'lightskyblue'],
```

```
        wedgeprops={'edgecolor': 'black'})
plt.title('Percentage of Sales by Product')
explode = (0,0,0,0.1)


plt.show()
```