**ASSIGNMENT 1 SOLUTIONS**

**QUESTION 1 :**

nums = [15, 8, 22, 7, 31, 4, 17]

print("Even numbers found in the list:")

for n in nums:

if n % 2 == 0:

print(n)

squared\_odds = []

for n in nums:

if n % 2 != 0:

squared\_odds.append(n \* n)

print("Squares of odd numbers:", squared\_odds)

output:

Even numbers found in the list:

8

22

4

Squares of odd numbers: [225, 49, 961, 289]

**QUESTION 2 :**

text = "The book was interesting because the book covered many topics and the topics discussed in the book were engaging"

words\_list = text.lower().split()

counts = {}

for word in words\_list:

counts[word] = counts.get(word, 0) + 1

print(counts)

output :

{'the': 4, 'book': 3, 'was': 2, 'interesting': 1, 'because': 1, 'covered': 1, 'many': 1, 'topics': 2, 'and': 1, 'discussed': 1, 'in': 1, 'were': 1, 'engaging': 1}

**QUESTION 3 :**

def check\_prime(number):

if number <= 1:

return False

for divisor in range(2, int(number \*\* 0.5) + 1):

if number % divisor == 0:

return False

return True

numbers\_list = [2, 4, 5, 10, 13, 17, 20, 23]

primes = [num for num in numbers\_list if check\_prime(num)]

print("Prime numbers:", primes)

output:

Prime numbers: [2, 5, 13, 17, 23]

**QUESTION 4 :**

import numpy as np

student\_scores = np.random.randint(50, 101, size=(10, 5))

avg\_scores = np.mean(student\_scores, axis=1)

max\_per\_subject = np.max(student\_scores, axis=0)

combined\_scores = np.column\_stack((student\_scores, avg\_scores))

print("Original Scores (10 students × 5 subjects):")

print(student\_scores)

print("\nAverage score per student:")

print(avg\_scores)

print("\nMaximum score per subject:")

print(max\_per\_subject)

print("\nFinal array with average score appended:")

print(combined\_scores)

**QUESTION 5 :**

# Define the header for the CSV file columns

header = "Subject1,Subject2,Subject3,Subject4,Subject5,Average"

# Save the final\_scores array to a CSV file with the header and 2 decimal places

np.savetxt("student\_scores.csv", final\_scores, delimiter=",", header=header, comments='', fmt='%.2f')

# Confirm that the file has been saved

print("File saved as student\_scores.csv")

**QUESTION 6 :**

import pandas as pd

data = {

"product": ["pen", "pencil", "notebook", "eraser", "marker", "scale", "sharpener", "glue"],

"Price": [1.5, 0.5, 2.0, 0.75, 1.8, 1.2, 0.9, 1.0],

"quantity": [10, 25, 15, 30, 12, 18, 20, 22]

}

df = pd.DataFrame(data)

print(df)

df['TotalValue'] = df['Price'] \* df['quantity']

print("\nAfter adding TotalValue column:")

print(df)

df = df.sort\_values(by='TotalValue', ascending=False)

print("\nSorted by TotalValue (descending):")

print(df)

avg\_quantity = df.groupby('Price')['quantity'].mean()

print("\nAverage quantity grouped by Price:")

print(avg\_quantity)

**QUESTION 7 :**

import pandas as pd

data = pd.read\_csv("train.csv")

print(data.isnull().sum())

print(data[data['Age'] < 18][['Name', 'Age']])

print(data[(data['Sex'] == 'female') & (data['Survived'] == 1)][['Name', 'Sex', 'Survived']])

print(data.groupby('Pclass')['Fare'].mean())

**#QUESTION 8 :**

pandas as pd

1. columns = ['Age', 'Sex', 'ChestPainType', 'RestingBP', 'Cholesterol',

'FastingBS', 'RestingECG', 'MaxHR', 'ExerciseAngina', 'Oldpeak',

'ST\_Slope', 'NumVesselsFluoro', 'Thalassemia', 'HeartDisease']

df = pd.read\_csv('processed.cleveland.data', names=columns)

2.import numpy as np

df.replace('?', np.nan, inplace=True)

df['NumVesselsFluoro'] = pd.to\_numeric(df['NumVesselsFluoro'], errors='coerce')

df['Thalassemia'] = pd.to\_numeric(df['Thalassemia'], errors='coerce')

df['Cholesterol'] = pd.to\_numeric(df['Cholesterol'], errors='coerce')

3.

a) print(df.isnull().sum())

b) df['NumVesselsFluoro'].fillna(df['NumVesselsFluoro'].median(), inplace=True)

df['Thalassemia'].fillna(df['Thalassemia'].mode()[0], inplace=True)

c)df.dropna(thresh=len(df.columns) - 2, inplace=True)

d) grouped = df.groupby('HeartDisease')['Cholesterol'].mean()

print(grouped)

**Question 9 :**

import matplotlib.pyplot as plt

students = ['Alic', 'Bob', 'Charlie', 'David', 'Eva']

math = [85, 78, 92, 88, 76]

science = [90, 82, 89, 94, 75]

english = [78, 85, 88, 80, 82]

history = [84, 80, 91, 79, 77]

plt.bar(students, math)

plt.title('Math Marks')

plt.xlabel('Students')

plt.ylabel('Marks')

plt.show()

plt.plot(students, math, label='Math')

plt.plot(students, science, label='Science')

plt.plot(students, english, label='English')

plt.plot(students, history, label='History')

plt.title('Marks in All Subjects')

plt.xlabel('Students')

plt.ylabel('Marks')

plt.legend()

plt.show()

**QUESTION 10 :**

def check\_bmi(weight, height):

bmi\_value = weight / (height \*\* 2)

bmi\_rounded = round(bmi\_value, 2)

if bmi\_value < 18.5:

status = "Underweight"

elif bmi\_value < 25:

status = "Normal"

elif bmi\_value < 30:

status = "Overweight"

else:

status = "Obese"

return f"BMI is {bmi\_rounded}: {status}"

print(check\_bmi(60, 1.7))

print(check\_bmi(80, 1.7))

print(check\_bmi(45, 1.6))

output:

BMI is 20.76: Normal

BMI is 27.68: Overweight

BMI is 17.58: Underweight