**1. Given a string, write a Python script to find the print the EVEN length words.**

def evenword(sentence):

word=sentence.split()

even=[]

for words in word:

if(len(words)%2==0):

even.append(words)

return even

sentence=input("Enter a sentence:")

even=evenword(sentence)

print("Words with even no of wordlength are:",even)

**2. Write a Python code to count number of substrings of a string.**

def substring(word):

n=len(word)

count=0

for i in range(n):

for j in range(i,n):

count+=1

return count

word=input("Enter a string:")

count=substring(word)

print("count of substring in given string are:",count)

**3. Python to program to extract the url from the given input string**

import re

def extracturl(string):

url=re.findall(r'\S+://\S+',string)

return url

string=input("Enter the string:")

url=extracturl(string)

print("The url is:",url)

**4. Write a Python program to calculate gross pay**

def grosspay(hours,payrate,ot,pay1):

gross=hours\*payrate

otpay=ot\*pay1

totalgross=gross+otpay

return totalgross

hours=int(input("Enter number of hours worked:"))

payrate=int(input("Enter payrate per hour:"))

ot=int(input("Enter number of hours worked over time:"))

pay1=int(input("Enter payrate per hour for overtime:"))

totalgross=grosspay(hours,payrate,ot,pay1)

print("Gross pay is:",totalgross)

**5. Python program for passing multiple arguments to a function**

def multiplearguments(\*argu):

for arg in argu:

print(arg)

multiplearguments("HEY",45689,[1,"SHREYAS",None,True])

def multiplearguments(arg1,arg2,arg3,arg4):

print(arg1,arg2,arg3,arg4)

arg1="STRING"

arg2=98

arg3=None

arg4=True

multiplearguments(arg1,arg2,arg3,arg4)

**6. Write a function named collatz() that has one parameter named number. If number is even, then collatz() should print number // 2 and return this value. If number is odd, then collatz() should print and return 3 \* number + 1. Then write a program that lets the user type in an integer and that keeps calling collatz() on that number until the function returns the value 1.**

def collatz(number):

if number%2==0:

result=number//2

print(result)

return result

else:

result=3\*number+1

print(result)

return result

def main():

try:

number= int(input("Enter an integer: "))

while number!= 1:

number= collatz(number)

except ValueError:

print("Please enter a valid integer.")

main()

**7. Fantasy Game Inventory: You are creating a fantasy video game. The data structure to model theplayer’s inventory will be a dictionary where the keys are string values describing the item in the inventory and the value is an integer value detail ing how many of that item the player has. Eg: The dictionary value {'rope': 1, 'torch': 6, 'gold coin': 42, 'dagger': 1, 'arrow': 12} means the player has 1 rope, 6 torches, 42 gold coins, and so on. Write a function named displayInventory() that would take any possible “inventory” and display it like the following: Inventory: 12 arrow 42 gold coin 1 rope 6 torch 1 dagger Total number of items: 62**

def displayInventory(inventory):

print("Inventory:")

total\_items = 0

for item, quantity in inventory.items():

print(str(quantity) + " " + item)

total\_items += quantity

print("Total number of items:", total\_items)

player\_inventory = {'rope': 1, 'torch': 6, 'gold coin': 42, 'dagger': 1, 'arrow': 12}

displayInventory(player\_inventory)

**8. Strong Password Detection: Write a function that uses regular expressions to make sure the password string it is passed is strong. A strong password is defined as one that is at least eight characters long, contains both uppercase and lowercase charac ters, and has at least one digit. You may need to test the string against mul tiple regex patterns to validate its strength.**

import re

def is\_strong\_password(password):

if len(password) < 8:

return False

if not re.search(r'[a-z]', password) or not re.search(r'[A-Z]', password):

return False

if not re.search(r'\d', password):

return False

return True

password=input("Enter the password:")

if is\_strong\_password(password):

print(f"'{password}' is a strong password.")

else:

print(f"'{password}' is not a strong password.")

1. **Create a Mad Libs program that reads in text files and lets the user add their own text anywhere the word ADJECTIVE, NOUN, ADVERB, or VERB appears in the text file.**

**Eg:A text file may look like this:**

**The ADJECTIVE panda walked to the NOUN and then VERB. A nearby NOUN was unaffected by these events.**

**The program would find these occurrences and prompt the user to replace them. Enter an adjective:**

**silly**

**Enter a noun: chandelier Enter a verb: screamed Enter a noun: pickup truck**

**The following text file would then be created:**

**The silly panda walked to the chandelier and then screamed. A nearby pickup truck was unaffected by these events.**

**The results should be printed to the screen and saved to a new text file.**

def process\_text(txt):

words = txt.split()

madlibs = []

for word in words:

if word == "ADJECTIVE":

usrinput = input("Enter an adjective: ")

madlibs.append(usrinput)

elif word == "VERB":

usrinput = input("Enter a verb: ")

madlibs.append(usrinput)

elif word == "NOUN":

usrinput = input("Enter a noun: ")

madlibs.append(usrinput)

else:

madlibs.append(word)

return ' '.join(madlibs)

def main():

input\_file\_path = input("Enter the path of the input text file: ")

output\_file\_path = input("Enter the path for the output text file: ")

with open(input\_file\_path, 'r') as input\_file:

intxt = input\_file.read()

processed\_text = process\_text(intxt)

print("\nProcessed Text:")

print(processed\_text)

with open(output\_file\_path, 'w') as output\_file:

output\_file.write(processed\_text)

print(f"\nProcessed text has been saved to '{output\_file\_path}'.")

main()

**10. Python program for Basic Shop Management System**

The program gets the product information from the user on product ID, product Name, product's Rate and product's Stock. Then ask user for updates on shop product for :

1.Show All Product

2.Search By ID

3.Search By Name

4.Sale The Product

5.Make Product Purchase

6.Exit System

**Note:** The above to be implemented in the lab and printout of the same along with output with soft bind need to submitted to Madhu by 28th August 2023 without fail.

**In the final lab exams for PART B questions:** 20 Marks is allotted.

class Product:

def \_init\_(self, product\_id, name, rate, stock):

self.product\_id = product\_id

self.name = name

self.rate = rate

self.stock = stock

class Shop:

def \_init\_(self):

self.products = []

def add\_product(self, product):

self.products.append(product)

def show\_all\_products(self):

for product in self.products:

print(f"ID: {product.product\_id}, Name: {product.name}, Rate: {product.rate}, Stock: {product.stock}")

def search\_by\_id(self, product\_id):

for product in self.products:

if product.product\_id == product\_id:

return product

return None

def search\_by\_name(self, name):

for product in self.products:

if product.name == name:

return product

return None

def sale\_product(self, product\_id, quantity):

product = self.search\_by\_id(product\_id)

if product:

if product.stock >= quantity:

product.stock -= quantity

print(f"Sold {quantity} units of {product.name}")

else:

print("Insufficient stock")

else:

print("Product not found")

def make\_purchase(self, product\_id, quantity):

product = self.search\_by\_id(product\_id)

if product:

product.stock += quantity

print(f"Purchased {quantity} units of {product.name}")

else:

print("Product not found")

def main():

shop = Shop()

while True:

print("1. Show All Product")

print("2. Search By ID")

print("3. Search By Name")

print("4. Sale The Product")

print("5. Make Product Purchase")

print("6. Exit System")

choice = int(input("Enter your choice: "))

match choice:

case 1:

shop.show\_all\_products()

case 2:

product\_id = int(input("Enter product ID: "))

product = shop.search\_by\_id(product\_id)

if product:

print(f"ID: {product.product\_id}, Name: {product.name}, Rate: {product.rate}, Stock: {product.stock}")

else:

print("Product not found")

case 3:

product\_name = input("Enter product name: ")

product = shop.search\_by\_name(product\_name)

if product:

print(f"ID: {product.product\_id}, Name: {product.name}, Rate: {product.rate}, Stock: {product.stock}")

else:

print("Product not found")

case 4:

product\_id = int(input("Enter product ID: "))

quantity = int(input("Enter quantity sold: "))

shop.sale\_product(product\_id, quantity)

case 5:

product\_id = int(input("Enter product ID: "))

quantity = int(input("Enter quantity purchased: "))

shop.make\_purchase(product\_id, quantity)

case 6:

break

case \_:

print("Invalid choice")

main()