2) (a+b+c) \* (a-b-c) \* a \* b + a\*\*2 + b\*\*2 + (a\*b\*c)\*\*3 🡪 equation

def create\_equation(a, b, c):

return (a+b+c) \* (a-b-c) \* a \* b + a\*\*2 + b\*\*2 + (a\*b\*c)\*\*3

print(create\_equation(1, 2, 3))

Explanation:

Creating a function using keyword def

\*\* 🡪 power function

The create\_equation() returns the value of arithmetic expressions.

3)Natural number:

Code:

def is\_natural\_number(n):

if isinstance(n, int) and n > 0:

return True

return False

# Get input from the user

n = int(input("Enter a number: "))

if is\_natural\_number(n):

print(f"{n} is a natural number.")

else:

print(f"{n} is not a natural number.")

Explanation:

Natural number means a positive whole numbers(1,2,3…)

Isinstance() 🡪it check whether the user input is integer character and the value is greater than 0

It returns true if the user input satisfies the above conditions.

4) Palindrome Porgram:

Code:

class Palindrome:

def \_\_init\_\_(self, word):

self.word = word

def is\_palindrome(self):

reverse = self.word[::-1]

if self.word == reverse:

return True

else:

return False

def main():

word = input("Enter a word: ")

p = Palindrome(word)

if p.is\_palindrome():

print(f"{word} is a palindrome.")

else:

print(f"{word} is not a palindrome.")

if \_\_name\_\_ == "\_\_main\_\_":

main()

Explanation:

If the original string and the reversed string of the original string is equal then it is said to be palindrome.

Here we use class Palindrome and an constructor class named \_\_init\_\_ it initializes the current value to the variable.

The reverse variable holds the reversed string to be checked.

5)GUI

CODE:

import tkinter as tk

root = tk.Tk()

selected = tk.StringVar()

option1 = tk.Radiobutton(root, text="Option 1", variable=selected, value="Option 1")

option2 = tk.Radiobutton(root, text="Option 2", variable=selected, value="Option 2")

option1.pack()

option2.pack()

root.mainloop()

Explanation:

Tkinter is one of the python package for graphical user interface

Root variable is the that holds the display area of the interface.

.pack() is used to load the variables into the interface.

6)Distance between two points:

Code:

import math

class Point:

def \_\_init\_\_(self, x, y):

self.x = x

self.y = y

def distance(p1, p2):

try:

distance = math.sqrt((p1.x - p2.x)\*\*2 + (p1.y - p2.y)\*\*2)

return distance

except ValueError:

return "Invalid input."

p1 = Point(1, 2)

p2 = Point(3, 4)

print(distance(p1, p2))

Explanation:  
Value to be substituted in the distance formula math.sqrt((p1.x - p2.x)\*\*2 + (p1.y - p2.y)\*\*2)

7) File Handlig

with open("hello.txt", "w") as f:

f.write("hello world")

print("Text written to file.")

Explanation:

open() is used to create or open a file named “hello.txt”

“w” refers to write

Write() is used to put the content in the file named hello.txt

8)Regular expression

Code:

import re

x = " 89e9jcd^o38829@3%3,/mkl$w1"

numbers = re.findall(r"\d", x)

result = "".join(numbers)

print(result)

Explanation:

Here ‘\d’ is an escape sequence that finds all the digits in the variable

9) mylist = ['tree', 'apple', 'mango', 'melon']

urlist = ['wood','knife','axe']

mylist + urlist\

10num = int(input("Enter the number: "))

# If given number is greater than 1

if num > 1:

# Iterate from 2 to n / 2

for i in range(2, int(num/2)+1):

# If num is divisible by any number between

# 2 and n / 2, it is not primeEnter the number: 13

13 is a prime number

if (num % i) == 0:

print(num, "is not a prime number")

break

else:

print(num, "is a prime number")

else:

print(num, "is not a prime number")