4.(i) Palindrome:-

```
def is_palindrome(n):
    index = 0
    while index < len(n)//2:
        if n[index] != n[-1-index]:
            return False
        index += 1
    return True
n = input("Enter a String:")
```

```
if is_palindrome(n):
    print("The given string is palindrome:")
else:
    print("The given string is not palindrome.")
```

o/p:-

```
Enter a string: raccare
The given string is a palindrome:
```

ii) area of circle

```
def area_of_circle(radius):
    import math
    return math.pi * radius**2

def main():
    radius = float(input("Enter radius of the circle:"))
    area = area_of_circle(radius)
    print(f"The area of circle is: {area:.2f}")
```

o/p:-

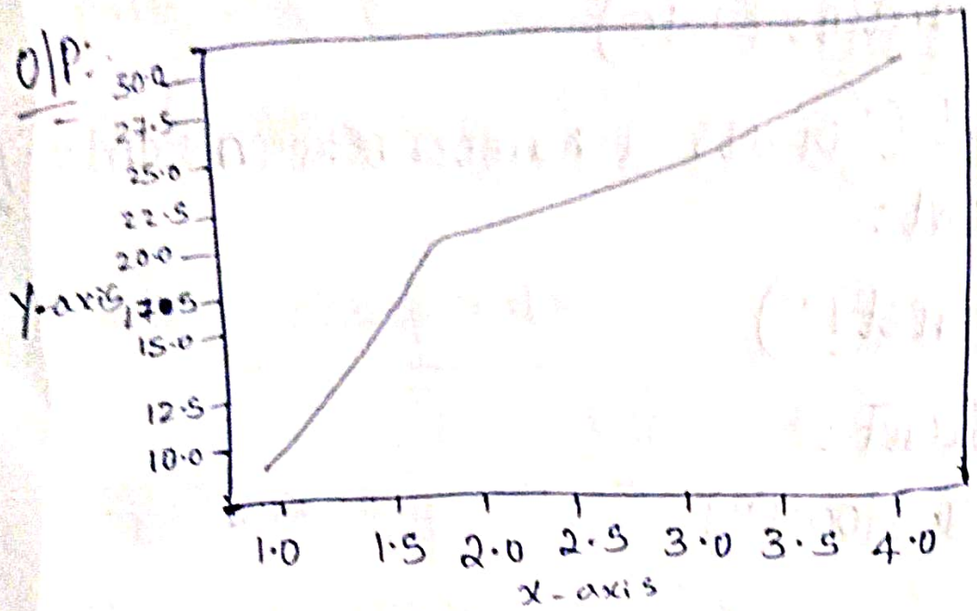
```
Enter radius of the circle: 5
The area of circle is : 78.54
```

5. Graph line:-

```
import matplotlib.pyplot as plt
x = [1,2,3,4]
y = [10,20,25,30]
```

```
plt.plot(x,y)
plt.title("Basic line Graph")
plt.xlabel("x-axis")
plt.ylabel("y-axis")
plt.show()
```

O/P:



b. soln: Try, Except, Finally + Errors

```
try:
    list = [1, 2, 3]
    Print (list[5])
except indexError :
    print ("Index out of range!")
Finally:
    Print ("Index Check completed.")


try:
    result = "2" + 2
except typeError:
```

print ("Type mismatch error!")

7. Guess number:.

```
import random
Secret = random.randint(1,10)
guess = int(input("Guess a number between 1 and
if guess == secret:
    print("correct!")
elif guess < secret:
    print("Too low!")
else:
    print("Too high!")
```

4. def check_string(s):
    if s == s[::-1]:
        print("palindrome")
    else:
        print("Not palindrome")

def area_of_circle(radius):
    return PI * radius * radius

def odd_even(n):
    if n%2==0:
        Print("Even")

```
else:
    print ("odd")

check_string("madam")
area_of_circle
print("Area:", area_circle(5))
odd_even(3)
```

8. Numpy Array Math

```
import numpy as np
arr = np.array([1,2,3])
print("original:", arr)
print("Add 2:", arr+2)
print("multiply by 3:", arr*3)
print("sum:", np.sum(arr))
print("mean:", np.mean(arr))
```

9. calculator & Search:-

```
a=10
b=5
print("Add:", a+b)
print("subtract:", a-b)
print("multiply:", a*b)
```

```
print("Divide:", a/b)

nums = [1, 2, 7, 9]
key = 7
found = False
for i in range(len(nums)):
    if nums[i] == key:
        print("Found at index:", i)
        found = True
        break
if not found:
    print("Not Found")
```

10. File copy & word count:

```
with open("source.txt", "w") as f:
    f.write("Hello world from python!")
with open("source.txt", "r") as f:
    content = f.read()
with open("copy.txt", "w") as f:
    f.write(content)
words = content.split()
print("Total words:", len(words))
```

11. Built-in, List, Tuple, Dict ops

```
print (len ("Hello"))
print (abs(-5))
print ( type(10))
print ( max(3,7))
Print (min(4,2))

my_list = [1,2,3]
my_list.append (4)
my_list.remove (2)
print (my_list [1])
my_list.insert (1,9)
print(my_list)

my_tuple = (1,2,3)
Print (my_tuple [0])
Print ( len( my_tuple))

my_dict = {"a": 1, "b": 2}
print (my_dict ["a"])
my_dict ["c"] = 3
Print (my_dict .get ("b"))
```

12. Lambda function:-

```
Square = lambda x: x*x
print ("square of 4 is:", Squar(4))
```

13. Matrix Addition & Transpose:-

```
import numpy as np
A = np.array([[1,2], [3,4]])
B = np.array([[5,6], [7,8]])
print("Matrix A:\n", A)
print("Matrix B:\n", B)
Sum_matrix = A+B
Transpose = A.T
print("sum:\n", Sum_matrix)
print("Transpose of A:\n", transpose)
```