1. Demonstrate Data Types

```
a = 10  # int
b = 10.5  # float
c = 2 + 3j  # complex
d = [1, 2, 3]  # list
e = (4, 5, 6)  # tuple
f = {"name": "John"}  # dictionary
g = True  # bool
h = b"bytes"  # bytes
i = "Hello"  # string
print(type(a), type(b), type(c), type(d), type(e), type(f), type(g), type(h), type(i))
```

2. Distance, Prime, Factorial

```
import math
x1, y1 = 0, 0
x2, y2 = 3, 4
distance = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
print("Distance:", distance)

start, end = 10, 20
print("Prime numbers:")
for num in range(start, end+1):
    if num > 1:
        for i in range(2, num):
            if num % i == 0:
                break
        else:
            print(num)

n = 5
fact = 1
for i in range(1, n+1):
    fact *= i
print("Factorial of", n, "is", fact)
```

3. Try, Except, Errors

```
try:
    my_list = [1, 2, 3]
    print(my_list[5])
except IndexError:
    print("Index out of range!")
finally:
    print("This is finally block.")

try:
    x = "hello" + 5
except TypeError:
    print("Type Error occurred!")
```

4. Number Guessing

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

5. Functions

```
def check_string(s):
    if s == s[::-1]:
        print("Palindrome")
    else:
        print("Not Palindrome")

def area_circle(radius):
    return 3.14 * radius * radius

def odd_even(n):
    if n % 2 == 0:
        print("Even")
    else:
        print("Odd")

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

6. 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("Mean:", np.mean(arr))
```

7. Calculator & Search

```
a = 10
b = 5
print("Add:", a + b)
print("Subtract:", a - b)
print("Multiply:", a * b)
print("Divide:", a / b)

nums = [4, 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")
```

8. 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))
```

9. 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"))
```

10. 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)
```

11. Lambda Function

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

12. String/List/Tuple Ops

```
s = "Hello World"
print("Slicing:", s[0:5])
print("Indexing:", s[6])
print("Formatted:", f"My string: {s}")

lst = [1, 2, 3]
lst[0], lst[-1] = lst[-1], lst[0]
print("Swapped List:", lst)
tup = tuple(lst)
print("Tuple:", tup)
new_list = list(tup)
print("List:", new_list)
```

13. Tkinter Message Box

```
import tkinter as tk
from tkinter import messagebox
def show_message():
    messagebox.showinfo("Message", "HELLO WORLD")
win = tk.Tk()
btn = tk.Button(win, text="Click Me", command=show_message)
btn.pack()
win.mainloop()
```

14. Matplotlib Line Graph

```
import matplotlib.pyplot as plt
x = [1, 2, 3]
y = [2, 4, 1]
plt.plot(x, y)
plt.xlabel("X Axis")
plt.ylabel("Y Axis")
plt.title("Simple Line Graph")
plt.show()
```

15. DataFrame using NumPy

```
import numpy as np
import pandas as pd
data = np.array([[1, 2], [3, 4]])
df = pd.DataFrame(data, columns=["A", "B"])
print(df)
```