1. Basic Datatypes in Python

```
python
CopyEdit
a = 10               # Integer
b = 3.14             # Float
c = "Hello"          # String
d = True             # Boolean
e = [1, 2, 3]        # List
f = (4, 5, 6)        # Tuple
g = {'x': 1, 'y': 2}  # Dictionary

print(type(a), type(b), type(c), type(d), type(e),
type(f), type(g))
```

---

2. Distance Between Two Points

```
python
CopyEdit
import math

x1, y1 = map(float, input("Enter x1 y1: ").split())
x2, y2 = map(float, input("Enter x2 y2: ").split())

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

---

3. a) Even Number Check

```
python
CopyEdit
num = int(input("Enter a number: "))
print("Even" if num % 2 == 0 else "Odd")
```

3. b) Decimal Equivalents using for loop

```
python
CopyEdit
for i in range(2, 11):
    print(f"1/{i} = {1/i}")
```

---

4. a) List and Tuple

```
python
CopyEdit
my_list = [10, 20, 30]
my_tuple = (40, 50, 60)
print("List:", my_list)
print("Tuple:", my_tuple)
```

4. b) For loop over a sequence

```
python
CopyEdit
sequence = ['a', 'b', 'c']
for item in sequence:
    print(item)
```

4. c) Countdown using while loop

```
python
CopyEdit
num = int(input("Enter a number: "))
while num >= 0:
    print(num)
    num -= 1
```

---

5. a) Sum of Primes below 2 million

```
python
CopyEdit
def is_prime(n):
    if n < 2: return False
    for i in range(2, int(n**0.5)+1):
        if n % i == 0:
            return False
    return True

print(sum(i for i in range(2, 2000000) if
is_prime(i)))
```

5. b) Sum of Even Fibonacci terms below 4 million

```
python
CopyEdit
a, b = 1, 2
total = 0
while b <= 4000000:
    if b % 2 == 0:
        total += b
    a, b = b, a + b
print(total)
```

---

6. a) Character Count in String

```
python
CopyEdit
s = input("Enter a string: ")
char_count = {}
for char in s:
    char_count[char] = char_count.get(char, 0) + 1
print(char_count)
```

6. b) Split, Join and Dictionary

```
python
CopyEdit
dob = "15/08/2000"
parts = dob.split("/")
dob_dict = {"day": parts[0], "month": parts[1],
"year": parts[2]}
joined = "-".join(parts)
print("Dictionary:", dob_dict)
print("Joined:", joined)
```

---

7. Frequency of Characters in a File

```
python
CopyEdit
filename = input("Enter file name: ")
with open(filename, 'r') as f:
```

```
    text = f.read()

freq = {}
for char in text:
    freq[char] = freq.get(char, 0) + 1
print(freq)

# Heuristic classification
if '#include' in text:
    print("Probably a C file")
elif 'def' in text or 'import' in text:
    print("Probably a Python file")
else:
    print("Probably a text file")
```

---

8. a) Print Each Line in Reverse

```
python
CopyEdit
filename = input("Enter file name: ")
with open(filename, 'r') as f:
    for line in f:
        print(line[::-1].strip())
```

8. b) Character, Word, Line Count

```
python
CopyEdit
filename = input("Enter file name: ")
with open(filename, 'r') as f:
    lines = f.readlines()

num_lines = len(lines)
num_words = sum(len(line.split()) for line in lines)
num_chars = sum(len(line) for line in lines)

print("Lines:", num_lines)
print("Words:", num_words)
print("Characters:", num_chars)
```

---

9. a) Nearly Equal Strings

```
python
CopyEdit
def nearly_equal(a, b):
    return sum(x != y for x, y in zip(a, b)) == 1 and
len(a) == len(b)

print(nearly_equal("hello", "hella"))
```

9. b) GCD and LCM in one line

```
python
CopyEdit
from math import gcd
lcm = lambda a, b: abs(a*b)//gcd(a, b)
print("GCD:", gcd(20, 30))
print("LCM:", lcm(20, 30))
```

---

10. a) Merge Sort

```
python
CopyEdit
def merge_sort(arr):
    if len(arr) > 1:
        mid = len(arr)//2
        L, R = arr[:mid], arr[mid:]
        merge_sort(L)
        merge_sort(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] < R[j]:
                arr[k] = L[i]; i += 1
            else:
                arr[k] = R[j]; j += 1
            k += 1
        while i < len(L): arr[k] = L[i]; i += 1; k +=
1
        while j < len(R): arr[k] = R[j]; j += 1; k +=
1
```

```
data = [38, 27, 43, 3, 9, 82, 10]
merge_sort(data)
print("Sorted:", data)
```

10. b) Insertion Sort and Selection Sort

```
python
CopyEdit
# Insertion Sort
def insertion_sort(arr):
    for i in range(1, len(arr)):
        key, j = arr[i], i-1
        while j >=0 and key < arr[j]:
            arr[j+1] = arr[j]
            j -= 1
        arr[j+1] = key

# Selection Sort
def selection_sort(arr):
    for i in range(len(arr)):
        min_idx = i
        for j in range(i+1, len(arr)):
            if arr[j] < arr[min_idx]:
                min_idx = j
        arr[i], arr[min_idx] = arr[min_idx], arr[i]

nums = [64, 25, 12, 22, 11]
insertion_sort(nums)
print("Insertion Sort:", nums)

nums2 = [64, 25, 12, 22, 11]
selection_sort(nums2)
print("Selection Sort:", nums2)
```