File no – 3:

**1. Check Palindrome without slicing**

👉 A palindrome reads the same forward and backward. Instead of slicing, compare characters from both ends moving inward.

def is\_palindrome(s):

for i in range(len(s)//2):

if s[i] != s[-i-1]:

return False

return True

**2. Remove duplicates from a list (without set)**

👉 Use a new list and check if an element already exists before appending. This keeps the original order.

def remove\_duplicates(lst):

result = []

for x in lst:

if x not in result:

result.append(x)

return result

**3. Merge two dictionaries**

👉 Copy one dictionary, then update it with the second. In case of duplicate keys, values from the second dictionary overwrite the first.

def merge\_dicts(d1, d2):

merged = d1.copy()

merged.update(d2)

return merged

**4. Character frequency in a string**

👉 Use a dictionary to count how many times each character appears. .get() helps handle missing keys.

def char\_frequency(s):

freq = {}

for ch in s:

freq[ch] = freq.get(ch, 0) + 1

return freq

**5. Implement your own max()**

👉 Initialize with the first element, then compare each remaining element, updating if a bigger one is found.

def my\_max(lst):

max\_val = lst[0]

for x in lst[1:]:

if x > max\_val:

max\_val = x

return max\_val