**Practical 6:**

**Q1.HINT**

**# Python code to count the number of occurrences**

**def countX(lst, x):**

**count = 0**

**for ele in lst:**

**if (ele == x):**

**count = count + 1**

**return count**

**# Driver Code**

**lst = [8, 6, 8, 10, 8, 20, 10, 8, 8]**

**x = 8**

**print('{} has occurred {} times'.format(x, countX(lst, x)))**

**Q2.**

**HINT**

**n = int(input("Input a number: "))**

**# use for loop to iterate 10 times**

**for i in range(1,11):**

**print(n,'x',i,'=',n\*i)**

**Q4.**

**HINT**

**1. ef garble(sentence):**

**return ' '.join([word[0] + ''.join(random.sample([char for char in word[1:-1]],len(word[1:-1]))) + word[-1] if len(word) > 1 else word for word in sentence.split()])**

**2.**

**def shuffle\_string(string):**

**chars = list(string)**

**random.shuffle(chars)**

**return ''.join(chars)**

**def garble\_word(word):**

**# No operation needed on sufficiently small words**

**# (Also, main algorithm requires word length >= 2)**

**if len(word) <= 3:**

**return word**

**# Split word into first & last letter, and middle letters**

**first, mid, last = word[0], word[1:-1], word[-1]**

**return first + shuffle\_string(mids) + last**

**def garble(sentence):**

**words = sentence.split(' ')**

**return ' '.join(map(garble\_word, words))**

**Practical 8**

**Q1.**

**HINT**

**def max\_min(data):**

**l = data[0]**

**s = data[0]**

**for num in data:**

**if num> l:**

**l = num**

**elif num< s:**

**s = num**

**return l, s**

**print(max\_min([0, 10, 15, 40, -5, 42, 17, 28, 75]))**

**Q2.HINT**

**def sum\_of\_cubes(n):**

**n -= 1**

**total = 0**

**while n > 0:**

**total += n \* n \* n**

**n -= 1**

**return total**

**print("Sum of cubes: ",sum\_of\_cubes(3))**

**Q3.HINT**

**def odd\_product(nums):**

**for i in range(len(nums)):**

**for j in range(len(nums)):**

**if i != j:**

**product = nums[i] \* nums[j]**

**if product & 1:**

**return True**

**return False**

**dt1 = [2, 4, 6, 8]**

**dt2 = [1, 6, 4, 7, 8]**

**print(dt1, odd\_product(dt1));**

**print(dt2, odd\_product(dt2));**