

1. Write a program to count the numbers of characters in the given string and store them in a dictionary data structure.

```
str=input("Enter a string:")
print("The number of characters is:", len(str))
dict={}
for c in str:
    dict[c]=str.count(c)
print("The configuration of the characters in the string is:")
print(dict)
```

✓ 2.2s

Python

```
The number of characters is: 5
The configuration of the characters in the string is:
{'f': 3, 'h': 2}
```

+ Code

+ Markdown

2. Write a program to use split and join methods in the given string and trace a birthday with a dictionary data structure.

```
n1=input("Enter name ")
a=input("Enter birthday in dd-mm-yy : ")
b=a.split("-")

n2=input("Enter name ")
x=input("Enter birthday in dd-mm-yy : ")
y=x.split("-")
ram={"Day":b[0],"Month":b[1],"Year":b[2]}
shyam={"Day":y[0],"Month":y[1],"Year":y[2]}
res1=n1+" born on " + a + " and " +n2+ " born on "+x
print(res1)

print(n1,"turns ",2024-int(ram["Year"])-2000," years on the
date",ram["Day"],ram["Month"],2024)
print(n2," turns ",2024-int(shyam["Year"])-2000," years on the
date",shyam["Day"],shyam["Month"],2024)
```

[18] ✓ 14.1s

Pyth

```
... Shreyaan borns on 07-01-05 and Shlok borns on 09-08-03
Shreyaan turns 19 years on the date 07 01 2024
Shlok turns 21 years on the date 09 08 2024
```

3. Write function to compute gcd and lcm of two numbers.

```
def gcd(a,b):
    if(b==0):
        return a
    else:
        return gcd(b,a%b)
def lcm(a,b):
    return a*b/gcd(a,b)
a=int(input("Enter the first number:"))
b=int(input("Enter the second number:"))
print("The GCD of the two numbers is:",gcd(a,b))
```

```
print("The HCF of the two numbers:", lcm(a,b))
```

```
... The GCD of the two numbers is: 5
    The HCF of the two numbers: 10.0
```

+ Code

+ Markdown

4. Write a function `ball_collide` that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding. Represent a ball on a plane as a tuple of (x, y, r), r being the radius. If (distance between two balls centers)  $\leq$  (sum of their radii) then (they are colliding)

```
a = tuple((input("Enter the parameters of the first ball: ").split()))
b = tuple((input("Enter the parameters of the second ball: ").split()))
def ball_collide(a,b):
    dist=((int(a[0])-int(b[0]))**2)+((int(a[1])-int(b[1]))**2)
    return dist == (int(a[2])+int(b[2]))**2

if ball_collide(a,b):
    print("They are colliding")
else:
    print("They are not colliding")
```

```
... They are colliding
```

5. Find mean, median, mode for the given set of numbers in a list.

```
import statistics
l = []
print("Enter number of elements :")
n = int(input())
for i in range(0, n):
    ele = int(input())
    l.append(ele)
print(l)
mean = statistics.mean(l)
median = statistics.median(l)
mode = statistics.mode(l)
print("The mean is:",mean)
print("The median is:",median)
print("The mode is:",mode)
```

```
Enter number of elements :
5
4
6
3
4
1
[4, 6, 3, 4, 1]
The mean is: 3.6
The median is: 4
The mode is: 4
ponk6745@Shreyaans-MacBook-Air python %
```

6. Write a program to implement

- a. Bubble sort:
- b. Merge sort
- c. Selection sort and
- d. Insertion sort.

Execute these sorting algorithms using switch case.

```
def bubbleSort(arr):  
  
    n = len(arr)  
  
    for i in range(n):  
  
        swapped = False  
  
        for j in range(0, n-i-1):  
  
            if arr[j] > arr[j+1]:  
  
                arr[j], arr[j+1] = arr[j+1], arr[j]  
  
                swapped = True  
  
        if not swapped:  
  
            break  
  
def merge(arr, l, m, r):  
  
    n1 = m - l + 1  
  
    n2 = r - m  
  
    L = [0] * n1  
  
    R = [0] * n2  
  
    for i in range(n1):  
  
        L[i] = arr[l + i]  
  
    for j in range(n2):  
  
        R[j] = arr[m + 1 + j]  
  
  
    i = 0  
  
    j = 0
```

```
k = 1

while i < n1 and j < n2:

    if L[i] <= R[j]:

        arr[k] = L[i]

        i += 1

    else:

        arr[k] = R[j]

        j += 1

    k += 1

while i < n1:

    arr[k] = L[i]

    i += 1

    k += 1

while j < n2:

    arr[k] = R[j]

    j += 1

    k += 1

def mergeSort(arr, l, r):

    if l < r:

        m = (l + r) // 2

        mergeSort(arr, l, m)

        mergeSort(arr, m + 1, r)

        merge(arr, l, m, r)
```

```

def selectionSort(arr):

    n = len(arr)

    for i in range(n):

        min_idx = i

        for j in range(i+1, n):

            if arr[j] < arr[min_idx]:

                min_idx = j

        arr[i], arr[min_idx] = arr[min_idx], arr[i]

def insertionSort(arr):

    n = len(arr)

    for i in range(1, n):

        key = arr[i]

        j = i - 1

        while j >= 0 and key < arr[j]:

            arr[j + 1] = arr[j]

            j -= 1

        arr[j + 1] = key

l = []

n = int(input("Enter number of elements: "))

for i in range(n):

    ele = int(input())

    l.append(ele)

print("Enter 1 for Bubble Sort, 2 for Merge Sort, 3 for Selection Sort, and 4 for Insertion Sort")

c = int(input())

```

```
match c:

    case 1:

        bubbleSort(l)

    case 2:

        mergeSort(l, 0, len(l) - 1)

    case 3:

        selectionSort(l)

    case 4:

        insertionSort(l)

print("Sorted list:")

for i in range(len(l)):

    print("%d" % l[i], end=" ")
```

```
ponk6745@Shreyaans-MacBook-Air python % /usr/local/bin/python3 /Users/ponk6745/Desktop/vscode/python/ass2.py
Enter number of elements : 5
2
3
4
5
1
Enter 1 for bubble sort, 2 for merge sort, 3 for selection sort and 4 for insertion sort
1
Sorted list:
1 2 3 4 5
ponk6745@Shreyaans-MacBook-Air python %
```