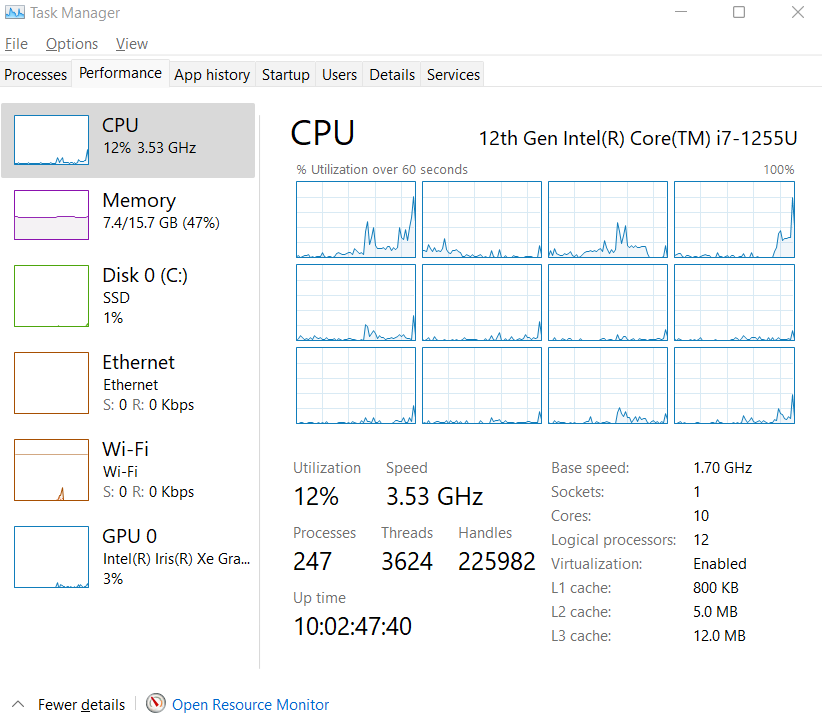
Saba Taha   
200901038  
BSCS-01-B

Operating Systems Assignment #03

Graphical user interface, text

Description automatically generated

**Code:**

import threading

#import random

import time

unsorted\_list = []

#for i in range (50):

# unsorted\_list.append((random.randint(0,100)))

size = int(input("Enter length of Array and then the Values: "))

for i in range (0, size):

elements = int(input())

unsorted\_list.append(elements)

print("Unsorted List: \n" + str(unsorted\_list) + "\n")

def merge(first\_half, second\_half):

i = j = 0

len\_f = len(first\_half)

len\_s = len(second\_half)

sorted\_list = []

while i<len\_f and j<len\_s:

if first\_half[i]<second\_half[j]:

sorted\_list.append(first\_half[i])

i+=1

else:

sorted\_list.append(second\_half[j])

j+=1

while i<=len\_f-1:

sorted\_list.append(first\_half[i])

i += 1

while j<=len\_s-1:

sorted\_list.append(second\_half[j])

j += 1

return sorted\_list

def merge\_sort(listx):

if len(listx) > 1:

mid = len(listx)//2

left\_side = listx[:mid]

right\_side = listx[mid:]

x = merge\_sort(left\_side)

y = merge\_sort(right\_side)

t1 = threading.Thread(target = x)

t2 = threading.Thread(target = y)

t1.start()

t2.start()

time.sleep(1.0)

t1.join()

t2.join()

return merge(x, y)

return listx

finished\_list = merge\_sort(unsorted\_list)

print("Sorted List Using Multithreaded MergeSort: \n" + str(finished\_list))