LP3 (DAA) Lab Exp No.3

def fractional\_knapsack(weights, values, capacity):

if len(weights) != len(values):

raise ValueError("Weights and values must be of the same length.")

res = 0

for pair in sorted(zip(weights, values), key=lambda x: x[1] / x[0], reverse=True):

if capacity <= 0:

break

if pair[0] > capacity:

res += capacity \* (pair[1] / pair[0])

capacity = 0

else:

res += pair[1]

capacity -= pair[0]

return res

def main():

num\_items = int(input("Enter the number of items: "))

weights = list(map(float, input("Enter the weights (space-separated): ").split()))

values = list(map(float, input("Enter the values (space-separated): ").split()))

if len(weights) != num\_items or len(values) != num\_items:

raise ValueError("The number of weights and values must match the number of items.")

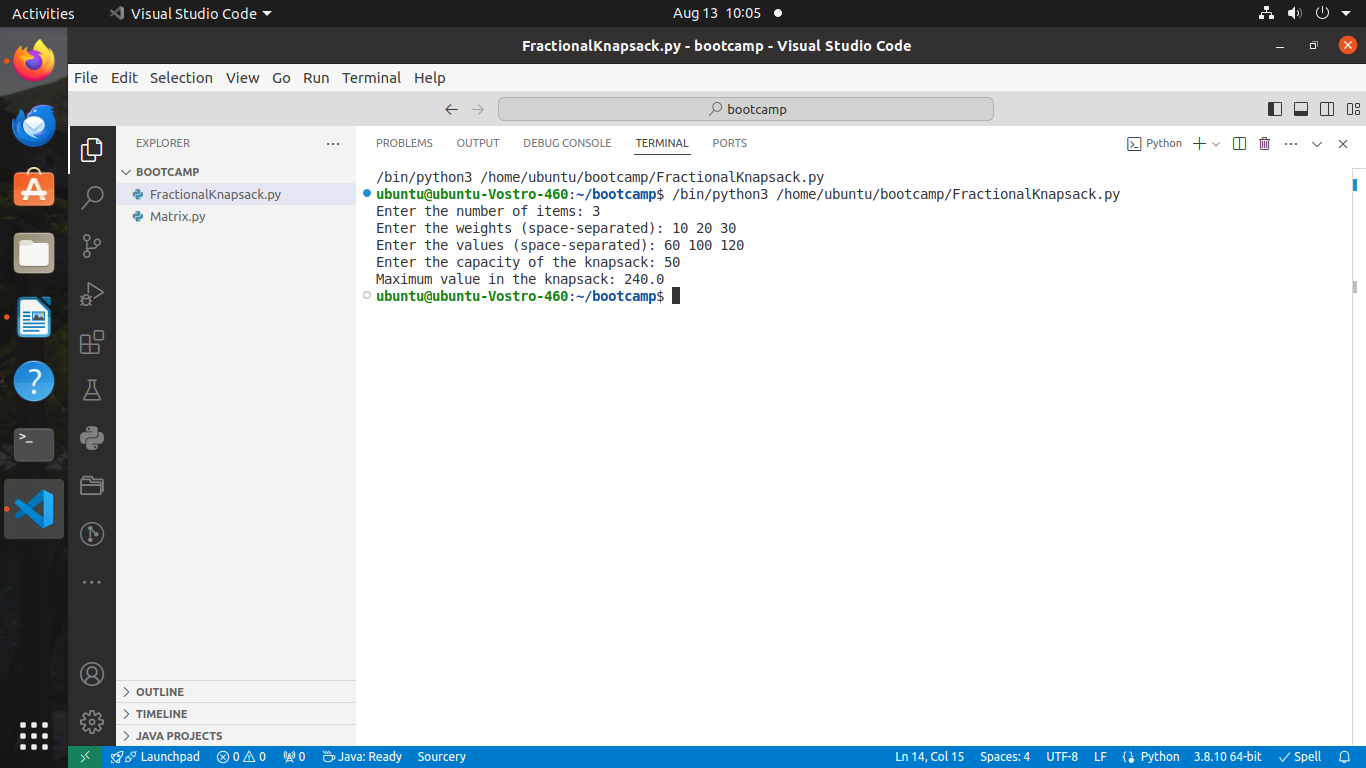
capacity = float(input("Enter the capacity of the knapsack: "))

max\_value = fractional\_knapsack(weights, values, capacity)

print(f"Maximum value in the knapsack: {max\_value}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

OUTPUT: -