112) Knapsack problem using greedy

CODE:

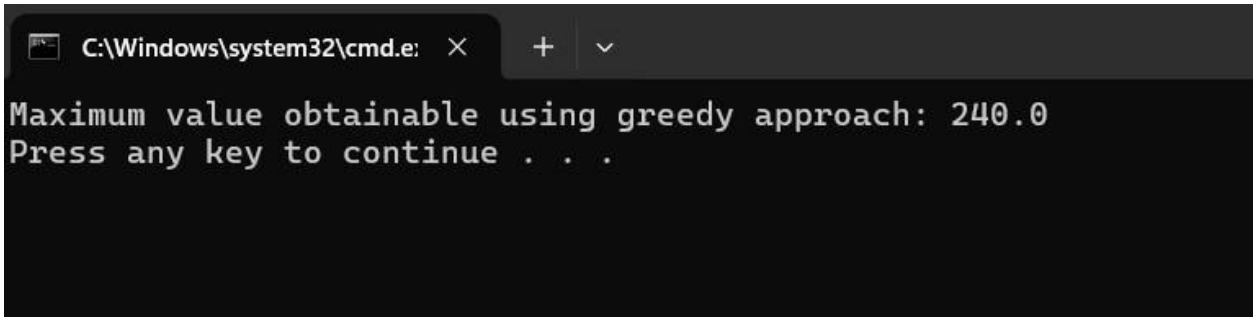
```
def knapsack_greedy(weights, values, capacity):
n = len(weights)
    value_per_weight = [(values[i] / weights[i], weights[i], values[i]) for i in
range(n)]
    value_per_weight.sort(reverse=True, key=lambda x: x[0])

    total_value = 0
total_weight = 0
        for ratio, weight, value in value_per_weight:
if total_weight + weight <= capacity:
total_value += value            total_weight += weight
else:           remaining_capacity = capacity -
total_weight            total_value += ratio *
remaining_capacity            break

    return total_value
 if __name__ ==
"__main__":     weights =
[10, 20, 30]     values =
[60, 100, 120]     capacity
= 50

    max_value = knapsack_greedy(weights, values, capacity)
    print(f"Maximum value obtainable using greedy approach: {max_value}")
```

OUTPUT:



TIME COMPLEXITY : O(n)