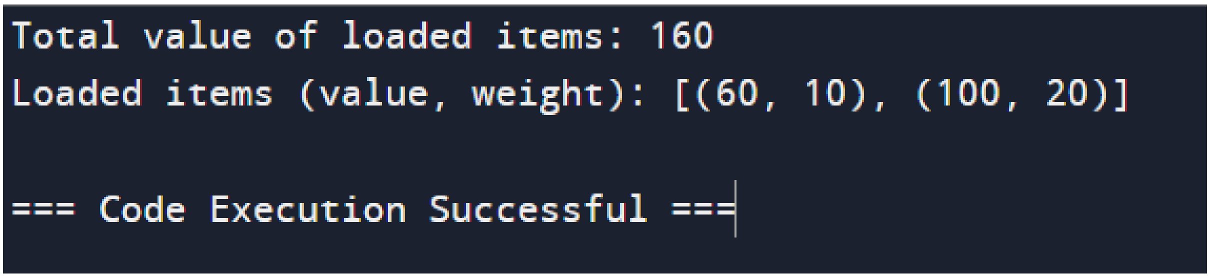
93. Container Loading ,

PROGRAM:-

```
def container_loading(weights, values, capacity):
    # Combine weights and values into a list of tuples (value, weight)
    items = list(zip(values, weights))

    # Sort items by value-to-weight ratio in descending order
    items.sort(key=lambda item: item[0] / item[1], reverse=True)

    total_value = 0  # Total value of items loaded into the container
    total_weight = 0  # Total weight of items loaded into the container
    loaded_items = []  # List of loaded items (value, weight)

    for value, weight in items:
        if total_weight + weight <= capacity:
            loaded_items.append((value, weight))
            total_weight += weight
            total_value += value
        else:
            break

    return total_value, loaded_items

# Example usage:
weights = [10, 20, 30]
values = [60, 100, 120]
capacity = 50

total_value, loaded_items = container_loading(weights, values, capacity)
print("Total value of loaded items:", total_value)
print("Loaded items (value, weight):", loaded_items)
```

OUTPUT:-



```
Total value of loaded items: 160
Loaded items (value, weight): [(60, 10), (100, 20)]

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n log n)