1. 144. You are given a list of items with their weights and values. Develop a program that utilizes exhaustive search to solve the 0-1 Knapsack Problem. The program should:
    1. Define a function total_value(items, values) that takes a list of selected items (represented by their indices) and the value list as input. It iterates through the selected items and calculates the total value by summing the corresponding values from the value list.

Code:

```
def total_value(selected_items, values):
    total = 0
    for item in selected_items:
        total += values[item]
    return total
from itertools import combinations

def knapsack_exhaustive_search(weights, values, capacity):
    num_items = len(weights)
    max_value = 0
    best_combination = None

    # Generate all possible combinations of items
    for r in range(num_items + 1):
        for combination in combinations(range(num_items), r):
            total_weight = sum(weights[i] for i in combination)
            if total_weight <= capacity:
                current_value = total_value(combination, values)
                if current_value > max_value:
                    max_value = current_value
                    best_combination = combination

    return best_combination, max_value

weights = [2, 3, 4, 5]
values = [3, 4, 5, 6]
capacity = 5

best_combination, max_value = knapsack_exhaustive_search(weights, values, capacity)
print("Best Combination:", best_combination)
print("Maximum Value:", max_value)
```

output:

Time complexity:f(n)=o(n!)