Assignment 4 –

public class Knapsack {

public static void main(String[] args) {

int[] weights = {2, 4, 6, 9};

int[] values = {10, 10, 12, 18};

int capacity = 15;

int[][] dp = knapsackDP(weights, values, capacity);

// Print the profit table

System.out.println("Profit Table:");

for (int i = 0; i <= weights.length; i++) {

for (int w = 0; w <= capacity; w++) {

System.out.print(dp[i][w] + "\t");

}

System.out.println();

}

int maxProfit = dp[weights.length][capacity];

System.out.println("Maximum Profit: " + maxProfit);

System.out.println("Selected items:");

int w = capacity;

for (int i = weights.length; i > 0 && maxProfit > 0; i--) {

if (dp[i][w] != dp[i - 1][w]) {

System.out.println("Item " + i + " (Weight: " + weights[i - 1] + ", Value: " + values[i - 1] + ")");

maxProfit -= values[i - 1];

w -= weights[i - 1];

}

}

}

public static int[][] knapsackDP(int[] weights, int[] values, int capacity) {

int n = weights.length;

int[][] dp = new int[n + 1][capacity + 1];

for (int i = 0; i <= n; i++) {

for (int w = 0; w <= capacity; w++) {

if (i == 0 || w == 0) {

dp[i][w] = 0;

} else if (weights[i - 1] <= w) {

dp[i][w] = Math.max(dp[i - 1][w], dp[i - 1][w - weights[i - 1]] + values[i - 1]);

} else {

dp[i][w] = dp[i - 1][w];

}

}

}

return dp;

}

}