**Assignment No: 4**

**Code:**

import java.util.Scanner;

public class KnapsackMain {

    // Function to solve the 0/1 Knapsack problem using dynamic programming

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

        int n = weights.length;

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

        // Build the dynamic programming table

        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], values[i - 1] + dp[i - 1][w - weights[i - 1]]);

                } else {

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

                }

            }

        }

        return dp[n][capacity];

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number of items: ");

        int n = scanner.nextInt();

        int[] values = new int[n];

        int[] weights = new int[n];

        System.out.println("Enter the values and weights for each item:");

        for (int i = 0; i < n; i++) {

            values[i] = scanner.nextInt();

            weights[i] = scanner.nextInt();

        }

        System.out.print("Enter the knapsack capacity: ");

        int capacity = scanner.nextInt();

        int maxProfit = knapsack(weights, values, capacity);

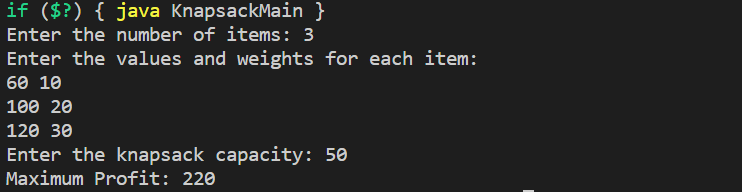
        System.out.println("Maximum Profit: " + maxProfit);

        scanner.close();

    }

}

**Output :**

****