public class Main {

public static void main(String[] args) {

int[] weights = {1, 3, 4, 5};

int[] profits = {1, 4, 5, 7};

int bagWeight = 8;

int maxProfit = knapsack(bagWeight, 0, weights, profits);

System.out.println("Maximum profit = " + maxProfit);

}

// 0/1 Knapsack recursive implementation

public static int knapsack(int bagWeight, int selectedElement, int[] weights, int[] profits) {

if (selectedElement >= weights.length || bagWeight == 0) {

return 0;

}

int pick = 0;

if (bagWeight >= weights[selectedElement]) {

pick = profits[selectedElement] + knapsack(bagWeight - weights[selectedElement], selectedElement + 1, weights, profits);

}

int notPick = knapsack(bagWeight, selectedElement + 1, weights, profits);

return Math.max(pick, notPick);

}

// Fibonacci using tabulation

public static long fibTabulation(int n) {

if (n <= 1) return n;

long[] memo = new long[n + 1];

memo[0] = 0;

memo[1] = 1;

for (int i = 2; i <= n; i++) {

memo[i] = memo[i - 1] + memo[i - 2];

}

return memo[n];

}

}