// DAA Practical 4 - 0/1 Knapsack Problem

import java.util.Arrays;

public class Main {

public static int knapsack(int[] value, int[] weight, int cap, int n) {

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

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

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

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

dp[i][w] = 0;

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

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

} else {

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

}

}

}

return dp[n][cap];

}

public static void main(String[] args) {

int[] value = {10, 5, 15};

int[] weight = {2, 3, 5};

int n = value.length;

int cap = 5;

int res = knapsack(value, weight, cap, n);

System.out.println(res);

}

}