import java.util.HashMap;

import java.util.Map;

import java.util.Objects;

public class KnapsackMemo {

static class Key {

int n, w;

Key(int n, int w) {

this.n = n;

this.w = w;

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (!(o instanceof Key)) return false;

Key key = (Key) o;

return n == key.n && w == key.w;

}

@Override

public int hashCode() {

return Objects.hash(n, w);

}

}

public static int knapsackMemo(int[] weight, int[] profit, int w, int n, Map<Key, Integer> memo) {

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

return 0;

}

Key key = new Key(n, w);

if (memo.containsKey(key)) {

return memo.get(key);

}

int result;

if (weight[n - 1] > w) {

result = knapsackMemo(weight, profit, w, n - 1, memo);

} else {

int include = profit[n - 1] + knapsackMemo(weight, profit, w - weight[n - 1], n - 1, memo);

int exclude = knapsackMemo(weight, profit, w, n - 1, memo);

result = Math.max(include, exclude);

}

memo.put(key, result);

return result;

}

public static void main(String[] args) {

int w1 = 4;

int[] profit1 = {1, 2, 3};

int[] weight1 = {4, 5, 1};

int n1 = profit1.length;

Map<Key, Integer> memo = new HashMap<>();

System.out.println("Output (Memoization): " + knapsackMemo(weight1, profit1, w1, n1, memo));

}

}