

1. Problem statement:

This is a program that solves the infamous knapsack problem, we are provided with a knapsack with limited space and a collection of items with different values and weights. The purpose is to maximize the value of items packed into the knapsack without exceeding the total weight capacity. For more details about the problem, check the Wikipedia description [from here](#)

2. Data format Specification:

A knapsack input contains $n + 1$ lines. The first line contains two integers, the first is the number of items in the problem, n . The second number is the capacity of the knapsack, K . The remaining lines present the data for each of the items. Each line, $i \in 0, \dots, n - 1$ contains two integers, the item's value V_i followed by its weight W_i .

Input Format

```
n K
v_0 w_0
v_1 w_1
...
v_{n-1} w_{n-1}
```

The output contains a knapsack solution and is made of two lines. The first line contains two values value and weight. Value is the total value of the items selected to go into the knapsack and weight is the total weight of the items selected. The next line is a list of n (0/1)-values (1 if the item is selected, 0 if not), one for each of the x_i variables. This line encodes the solution.