02 - 0-1 knapsack

We have a knapsack of volume V and n types of items, (i=0,1,...,n-1). Items of type i have weight w_i and volume v_i. The objective is to determine how many items of each type should be placed in the knapsack so as to maximize the total weight of the knpsack without exceeding its volume (V). At most one item of each type can be selected.

For instance, if we have a knapsack of volume V=1000 and n=5 types of items, i_0 (weight 40 and volume 200), i_1 (weight 50 and volume 314), i_2 (weight 100 and volume 198), i_3 (weight 95 and volume 500), and i_4 (weight 30 and volume 300). The most valuable knapsack not exceeding the volume limit contains only the items i_0, i_2, and i_3.

knapsack.hh

```
typedef struct {
        long unsigned int weight;
        long unsigned int volume;
} item;
/**
 * @param output The output array of item quantities
 * @param capacity The knapsack capacity
 * @param n The number of available items
 * @param items The array of n available items
 */
void knapsack(long unsigned int* output,const long unsigned int capacity,const long unsigned int n,const item* items);
```