

Large Knapsack

The [knapsack problem](#) or **rucksack problem** is a problem in combinatorial optimization: Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible. It derives its name from the problem faced by someone who is constrained by a fixed-size knapsack and must fill it with the most valuable items.

Just implement 0/1 Knapsack.

Input

First line contains two integers K and N, where K is the maximum knapsack size and N is the number of items. N lines follow where i^{th} line describes i^{th} item in the form v_i and w_i where v_i is the value and w_i is the weight of i^{th} item.

Output

Output a single number - maximum value of knapsack. (All operations and the answer are guaranteed to fit in signed 32-bit integer.)

Time limit changed to 2s on 02.07.11.

Example

Input:

```
10 3
7 3
8 8
4 6
```

Output:

```
11
```

Constraints:

$K \leq 2000000$

$N \leq 500$

$V_i \leq 10^7$

$W_i \leq 10^7$