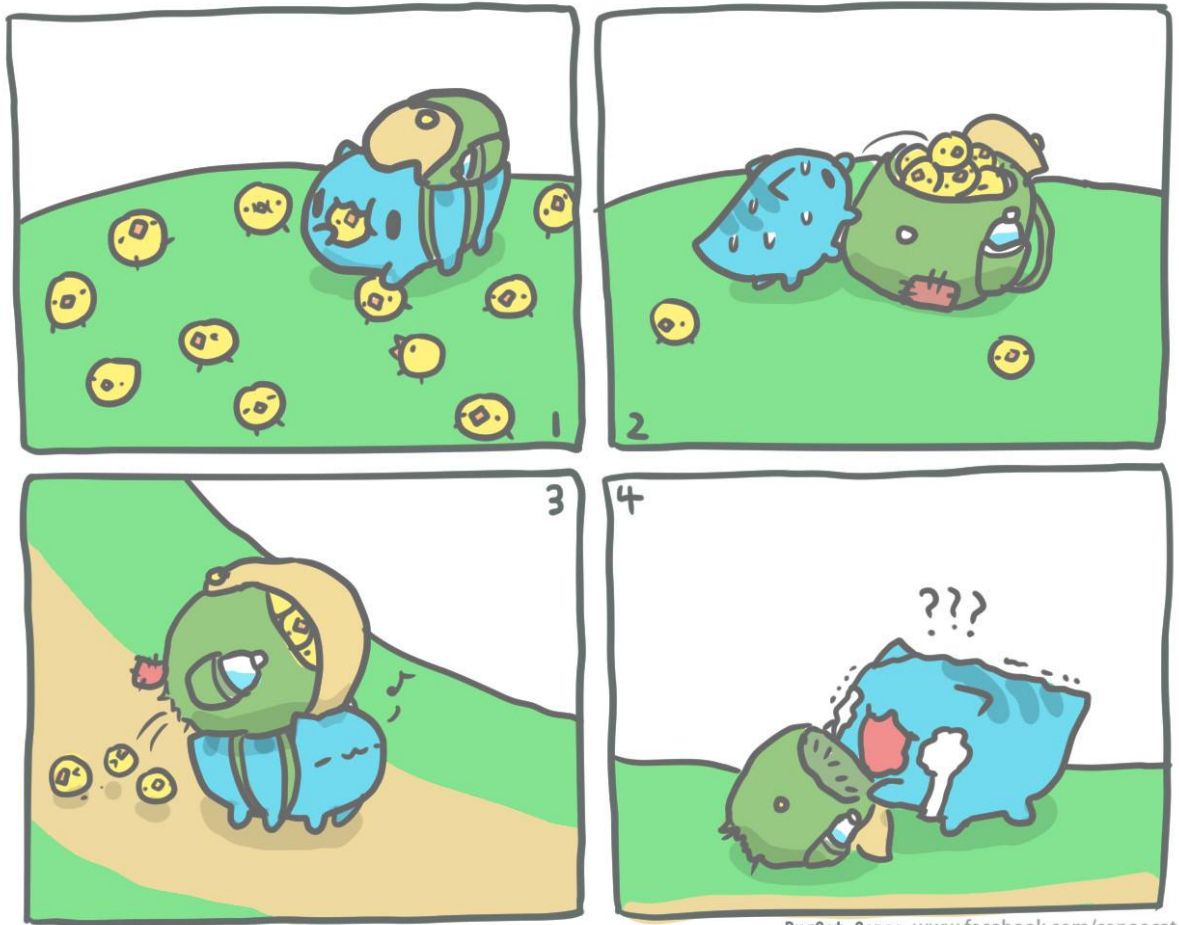


F. 不貳過

knapsack



吃了好多小雞之後，咖波想把小雞帶回家繼續享用。但是咖波想起了以前的慘痛教訓。

聰明的咖波不會犯同一種錯誤。他決定寫一個程式好好計算包包的大小。

咖波總共帶了 N 個包包，每個包包有不同的容量，每隻小雞也有不同的大小和飽足度。

比較大隻的小雞不代表飽足度就愈高喔！

咖波想要他帶回家的所有小雞的飽足度總和能夠最大，你能寫個程式幫他算一算嗎？

輸入說明

第一行有一個正整數 N ，代表咖波帶了 N 個包包。

第二行有 N 個正整數， $B_0, B_1, B_2, \dots, B_{N-1}$ ，代表這 N 個包包的大小，大小為 B_i 的包包裡面裝的小雞大小總和不得超過 B_i ，否則就會發生和上次一樣的慘劇喔！

第三行有一個正整數 M ，代表有 M 隻小雞。

接下來 M 行，每行兩個數字， W_i, V_i ，分別代表第 i 隻小雞的重量和飽足度。

輸出說明

輸出一個正整數，代表帶回去的小雞飽足度總和最多是多少？

輸入限制

$$1 \leq N \leq 3$$

$$1 \leq B_i \leq 70$$

$$1 \leq M \leq 100$$

$$1 \leq W_i \leq 100$$

$$1 \leq V_i \leq 100$$

子任務

子任務	分數	額外輸入限制
1	102	$N = 1$
2	99	$N = 2$
3	99	$N = 3$

範例一

INPUT	OUTPUT
2 10 7 3 6 10 5 4 8 6	16

範例二

INPUT	OUTPUT
1 12 3 6 10 5 4 8 6	14