**0 - 1 Knapsack Problem:-**

You are given weights and values of **N** items, put these items in a knapsack of capacity **W** to get the maximum total value in the knapsack. Note that we have only **one quantity of each item**.  
In other words, given two integer arrays **val[0..N-1]** and **wt[0..N-1]** which represent values and weights associated with **N** items respectively. Also given an integer W which represents knapsack capacity, find out the maximum value subset of **val[]** such that sum of the weights of this subset is smaller than or equal to **W.** You cannot break an item, **either pick the complete item, or don’t pick it (0-1 property)**.

**Input:**  
The first line of input contains an integer **T** denoting the number of test cases. Then **T** test cases follow. Each test case consists of four lines.  
The first line consists of **N** the number of items.  
The second line consists of **W**, the maximum capacity of the knapsack.  
In the next line are **N** space separated positive integers denoting the values of the **N** items,  
and in the fourth line are **N** space separated positive integers denoting the weights of the corresponding items.

**Output:**  
For each testcase, in a new line, print the **maximum possible** value you can get with the given conditions that you can obtain for each test case in a new line.

**Constraints:**  
1 ≤ T ≤ 100  
1 ≤ N ≤ 1000  
1 ≤ W ≤ 1000  
1 ≤ wt[i] ≤ 1000  
1 ≤ v[i] ≤ 1000

**Example:**  
**Input:**  
2  
3  
4  
1 2 3  
4 5 1  
3  
3  
1 2 3  
4 5 6  
**Output:**  
3  
0  
**Explanation:**  
**Test Case 1:**With W = 4, you can either choose the 0th item or the 2nd item. Thus, the maximum value you can generate is the max of val[0] and val[2], which is equal to 3.

**Test Case 2:**With W = 3, there is no item you can choose from the given list as all the items have weight greater than W. Thus, the maximum value you can generate is 0.