The Avengers

The avengers are a team of superheroes defending Earth against super villains and invading aliens and other mega threats. This team includes members like: Tony Stark (Iron Man), Steve Rogers (Captain America), Natasha Romanoff (Black Widow), Bruce Banner (The Hulk) and Thor among others.

Earth is now facing a new threat, the mad titan: Thanos, who is in search for the extremely powerful infinity stones and once he gets them, life as we know it will end!

The avengers are earth's last hope against Thanos. However, to have a chance against the titan they need special powerful weapons made of the strongest materials. And this is where we introduce Vibranium, a rare secret metal of extraterrestrial origin that is almost indestructible (probably the only equally strong metal in existence is Adamantium which exists in the hero Wolverine's body). An example of a weapon made with Vibranium is Captain America's Shield.

The problem is that Vibranium is a very rare metal and it only exists in Wakana (a country in Africa). Luckily, the king of Wakanda T'Challa (Black Panther) is a member of the Avengers and so he is ready to offer them all the Vibranium reserves they have. In addition to that, King T'Challa will utilize the advanced Wakandian technology to create Vibranium weapons. Using this technology Stark Industries was able to create special Vibranium weapons. These weapons come in special containers. Each of these containers has an **unbounded number of weapons** of different types (you can take as much weapons from any container as you want). Each weapon inside the container has a cost (how much Vibranium is required for its creation) and a power rating associated with it (how powerful the weapon is). The Avengers have to decide wisely on what weapons to create so as to maximize the combined total power of all the weapons while using as little Vibranium as possible. Note that the container itself is made of Vibranium and it costs a certain amount of it to be created.

Input

c w v (c: no of containers, w: no of weapons in each container, v : total amount of Vibranium available for creating the weapons)

c intergers specifying cost of creating containers (from container 1 to container c)
c*w numbers specifying cost of weapon j from container i
c*w numbers denoting the rating of weapon j from container i

Output

The maximum total possible power of the weapons

Example 1 (in this example there is only one weapon container).

Input

```
1 4 27

4

3 1 3 5

4 2 6 5
```

In the above example, there is one container, and inside it 4 weapons and a total of 27 Vibranium units available to us. The container itself costs 4 units of Vibranium to be made. The container comes with the following possible weapons: first weapon costs 3 units and has a power rating of 4. The 2nd weapon costs 1 units and has a power rating of 2. The 3rd weapon costs 3 units of Vibranium and has a power rating of 6. The 4th weapon costs 5 units and has a power rating of 5.

Output

```
46
```

The results is 46, i.e., the maximum possible weapon power we could gain is 46. This achieved by creating 7 copies of weapon 3 and 2 copies of weapon 2. So we get a total power of 7*6 + 2*2 = 46. Note that the cost for this is 7*3 + 2*1 + 4 (container 1 cost) = 27 Vibranium units.

Note that in the above example there is only one container. In other examples, there might be 2 or more containers. In the case of more than one container, you have to option to create weapons from any container or have a group of weapons from a certain container and another group from another one. However, if you decide to create even one weapon from a certain container you have to add the price for that container.

Example 2 (In this example there are two weapon containers).

Input

```
2 4 25

2 1

5 3 4 4

2 1 3 5

2 9 3 7

5 2 6 5
```

Output

```
65
```

The results is 65, i.e., the maximum possible weapon power we could gain is 65. This achieved by creating 7 copies of weapon 2 from the first container and 1 copy of weapon 2 of the 2nd container. So we get a total power of 7*9 + 1*2 = 65. Note that the cost for this is 7*3 + 1*1 + 2 (container 1 cost) + 1 (container 2 cost) = 25 Vibranium units.

The Project is two parts:

1- In the first part all the test cases have 1 container. For this part you have to submit the dynamic programming solution on the hackerrank.

2- In the second part all the test cases have 2 or more containers. This part is more challenging that the first part. For this part you only have to submit the divide and conquer solution only on the hackerrank.

The first part has 80% of the mark and the 2nd part has 20% of the mark.

Here is a reminder of the outline of your dynamic programming solution:

Step 1: Divide & Conquer

1- Define the value returned by the function *f* which we want to optimize.

2- Define the parameters which *f* depends on.

3- Draw the **recursion tree** for *f* using the values from the example above.

4- Write the recursive (divide and conquer) code to solve the question.

Step 2: Dynamic Programming

5- Draw the table and determine the dependencies between the table cells.

6- Determine the direction of movement within the table.

7- Write the Dynamic programming code which fills the table(s).

8- Write the code that will print the sequence of moves that go you the solution.

You are requested to submit a report that explains each of the steps above which also includes graphs and figures to explain your solution and the rationale behind it. Also, you need to submit your working code on the hackerrank website as usual. The link will be provided on Moodle/FB Group.

Important:

To solve this task you are not allowed to copy/be inspired by any piece of code from the internet or from a colleague or from anyone or any place.

If any percentage of resemblance is found between your code and a code listed on the internet (even if the code is a solution for a different problem), it will be considered cheating.

You are only allowed to check the code of the three problems we studied at class and the last years problems I posted on Moodle.

If it has been proven that you cheated on this task (no matter how small the percentage is), you will get zero in the final exam mark.