

## JamP5: Assemble

### Time limit: 4 seconds

Recently your team noticed that the computer you use to practice for programming contests is not good enough anymore. Therefore, you decide to buy a new computer. To make the ideal computer for your needs, you decide to buy separate components and assemble the computer yourself. You need to buy exactly one of each type of component.

The problem is which components to buy. As you all know, the quality of a computer is equal to the quality of its weakest component. Therefore, you want to maximize the quality of the component with the lowest quality, while not exceeding your budget.

### Input

On the first line one positive number: the number of testcases, at most 100. After that per testcase:

- One line with two integers:  $1 \leq n \leq 1000$ , the number of available components and  $1 \leq b \leq 1000000000$ , your budget.
- $n$  lines in the following format: “**type name price quality**”, where type is a string with the type of the component, name is a string with the unique name of the component, price is an integer ( $0 \leq \text{price} < 1000000$ ) which represents the price of the component and quality is an integer ( $0 \leq \text{quality} \leq 1000000000$ ) which represents the quality of the component (higher is better). The strings contain only letters, digits and underscores and have a maximal length of 20 characters.

It will always possible to construct a computer with your budget.

### Output

Per testcase: One line with one integer, the maximal possible quality.

### Sample Input

```
1
18 800
processor 3500_MHz 66 5
processor 4200_MHz 103 7
```





processor 5000\_MHz 156 9  
processor 6000\_MHz 219 12  
memory 1\_GB 35 3  
memory 2\_GB 88 6  
memory 4\_GB 170 12  
mainbord all\_onboard 52 10  
harddisk 250\_GB 54 10  
harddisk 500\_FB 99 12  
casing midi 36 10  
monitor 17\_inch 157 5  
monitor 19\_inch 175 7  
monitor 20\_inch 210 9  
monitor 22\_inch 293 12  
mouse cordless\_optical 18 12  
mouse microsoft 30 9  
keyboard office 4 10

### sample output

9

