Assembly line

The Alternate Control Machine (ACM) Factory has a large assembly line to make a type of product. The assembly has N robots (R_1, R_2...R_N) working sequentially. That means a semi-finished product moves from robot R_1, to R_2, then to R_3... to R_N. Each robot adds a component to the product. Each robot can complete its own job in P_i products per one hour.

The company has a budget of M VNĐ to improve productivity for the entire assembly. As a product director, you know that robot R_i needs to invest M_i VNĐ to contribute to the production of one more product per hour. You have to optimize the amount of money to invest to each robot to produce maximum number of products per hour.

Input

The first line of input contains one integer T ($1 \le T \le 10$) - the number of test cases.

Then T test cases are given as follows:

- The first line of each test case contains an integer N ($1 \le n \le 10^5$) and an integer M ($0 \le M \le 10^{12}$) – the number of robots and the budget
- Line i-th of the next N lines contains two integers P_i ($1 \le P_i \le 10^9$) and M_i ($1 \le M_i \le 10^9$) – information of the robot i-th

Output

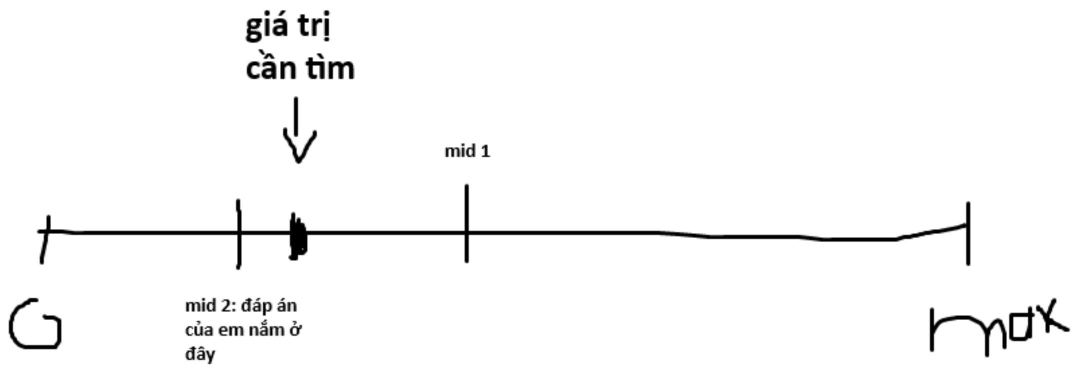
For each test case, output in one line the maximum number of products the assembly can make after investing at most M VNĐ.

Example

Input:
```
1
3 7
1 2
2 3
3 1
```

Output:
```
3
```

Hint: You should check for the case: company budget is 0, or cannot upgrade any robot at all. This is the visualization explain why you should check if you use binary search to solve.



budget = 0 => nếu ở mid2, sẽ có số tiền cần upgrade = 0 => em break => em trả về mid2 => sai vì bé hơn giá trị cần tìm