Maximum Skill 2

Accuracy: **47.37%** Submissions: **23K+** Points: **20**

Geek currently has skill **s** and **d** days left for placements. He has a array **problems** of **n** problems where i^{th} problem is represented as **problems[i]={rating$_i$, gain$_i$}.** Geek could solve the i^{th} problem only if **his currenty skill is not smaller than the rating$_i$**. And solving i^{th} problem will increase geek's skill by the **gain$_i$.**

Geek wants his skill to get maximised because the placement season is near. Find the **maximum rating Geek could reach** if he could solve **only one problem in a day, if there are problems that he could solve.**

Example 1:

Input:
s = 25
n = 3
d = 3
problems = {{35, 45}, {13, 6}, {100, 4}}
Output:
31
Explanation:
Geek could solve the second question on first day making his skill 31, he could not solve any problem on second and third days.

Example 2:

Input:
s = 5
n = 5
d = 3
problems = {{4, 3}, {17, 3}, {3, 5}, {9, 23}, {5, 1}}
Output:
36
Explanation:

Geek solves third question on first day, solves fouth question on second day and solves second question third day.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **maximumSkill2()** which takes integers **s, n, d and 2D integer array problems[][]** as input parameters and returns the maximum rating Geek could reach.

Constraints:

$1 <= s <= 10^5$

$1 <= d <= n <= 10^5$

$1 <= rating[i] <= 10^3$

$1 <= gain[i] <= 10^3$

C++ (g++ 5.4)

```
1   // } Driver Code Ends
29
30  class Solution {
31    public:
32      int maximumSkill2(int s, int n, int d, vector<vector<int>> &problems) {
33          // code here
34      }
35  };
36
37  // } Driver Code Ends
```

Custom Input | Compile & Run | Submit

Report An Issue

If you are facing any issue on this page. Please let us know.

Menu