UCF Local Contest (Final Round) — August 31, 2024

Team Work

filename: `teamwork`
Difficulty Level: `Hard`
Time Limit: `1 second`

The success of the UCF Programming Teams over 40+ years are attributed to the talented and dedicated students and to the experienced and great coaches. The coaches teach the various algorithms and team strategies, and the students deliver in various competitions.

The Problem:

A team in the International Collegiate Programming Contest consists of three students. One of the major team strategies is "who should do what problem". Wouldn't it be nice if the coaches had a program to answer this? And, wouldn't it be even nicer if you wrote the program for the coaches?!

Given the following information:

- number of students on a team,
- number of problems in the set,
- who can solve what problems and how long it takes to solve each problem, and
- total time available (how long the contest is),

determine the maximum number of problems that can be solved by the entire group in the given time. To simplify the problem, assume that:

- A student is limited to solving at most one problem in the entire contest. For example, let's assume the contest is 5 hours and Student_1 can solve Problem_7 in 1 hour and Problem_9 in 2 hours. Even though this student can finish both problems in 3 hours (within the 5-hour contest), the student is still limited to at most one problem in the entire contest.

- There is only one computer for the entire team. So, if a student is using the computer, the teammates cannot use the computer, i.e., problems must be solved one at a time. For example, let's assume the contest is 5 hours, Student_2 can solve Problem_6 in 2 hours and Student_3 can solve Problem_8 in 4 hours. Since problems are solved on the computer one at a time, the 5-hour contest is not enough to finish both problems since the two problems require 6 hours. Again, the problems are not solved in parallel even though there are multiple students; the problems are solved one at a time.

The Input:

The first input line contains three integers as follows:

- n ($1 \le n \le 200$), number of students on the team,
- p ($1 \le p \le 200$), number of problems in the set,
- t ($1 \le t \le 10^5$), total time available (how long the contest is).

The next *n* input lines provide who can solve what problems and how long it takes for each problem. Each of these *n* input lines contains *p* integers; row_1 is the information for $student_1$, row_2 is for $student_2$, and so on. The first integer on a row shows how long it takes that student to solve $problem_1$, the second integer on a row shows how long it takes that student to solve $problem_2$, and so on. Each of these integers are between 0 and 10^5, inclusive. A value of zero indicates that student can't solve that problem (doesn't know how to solve that problem).

The Output:

Print the maximum number of problems that can be solved by the entire group in the given time.

Sample Input	**Sample Output**
2 4 5 2 5 3 4 0 4 5 0	1
2 4 5 1 5 1 1 2 1 0 1	2