

GD6: First-Come-First-Serve with deadline (6 sec)

[Optional]

問題描述：

There are N jobs to be processed, where $N < 100,000$, and these jobs must be completed before a given deadline. The CEO wants to buy more machines to meet the deadline requirement. A job cannot be partitioned and must be processed in one machine. The i -th job takes $t(i)$ time no matter which machine you choose to process it. An important principle is first-come-first-serve. Suppose that the jobs have been sorted by their incoming time. For $i < j$, job j cannot start before the time that job i starts. Given a deadline, write a program to compute the minimum number of machines to complete all the jobs in time.

輸入說明：

The input consists of a number of test cases. The first line is an integer T which is the number of test cases, and the test cases follow one by one. The input of a test case consists of two lines. The first line contains two integers N and D , which are the number of jobs and the deadline, respectively. The second line consists of N integers, which are $t(0), t(1), \dots, t(N-1)$. Any two consecutive numbers in the same line are separated by a space. You can assume that all the input and output numbers in this problem are 32-bit integers.

輸出說明：

Output the minimum number of machines to complete all the jobs in time in one line. If it is impossible, output -1.

範例：

Sample Input:	Sample Output:
2	2
5 8	-1
2 3 1 5 4	
4 5	
5 5 2 6 4	