The 2009 ACM Asia Programming Contest Wuhan Site
sponsored by IBM
hosted by Wuhan University



Problem D
Download Manager
Input: download.in

Jiajia downloads a lot, a lot more than you can even imagine. Some say that he starts downloading up to 20,000 files together. If 20,000 files try to share a limited bandwidth then it will be a big hazard and no files will be downloaded properly. That is why, he uses a download manager.

If there are *T* files to download, the download manger uses the following policy while downloading files:

1. The download manager gives the smaller files higher priority, so it starts downloading the smallest *n* files at startup. If there is a tie, download manager chooses the one with less bytes remaining (for download). We assume that with at least 50 Mega Bytes/sec of bandwidth, *n* files can be downloaded simultaneously without any problem.

2. The available bandwidth is equally shared by the all the files that are being downloaded. When a file is completely downloaded its bandwidth is instantaneously given to the next file. If there are no more files left except the files that are being downloaded, this bandwidth is immediately shared equally by all remaining files that are being downloaded.

Given the size and completed percentage of each file, your task is to intelligently simulate the behavior of the download manager to find the total time required to download all the files.

Input
The will be at most 10 test cases. Each case begins with three integers *T* ($1 \le T \le 20000$), *n* ($1 \le n \le 2000$ and $1 \le n \le T$) and *B* ($50 \le B \le 1000$). Here *B* denotes the total bandwidth available to Jiajia (In Megabytes/sec). Please note that the download manager always downloads *n* files in parallel unless there are less than *n* files available for download. Each of next *T* lines contains one non-negative floating-point number *S* (less than 20,000, containing at most 2 digits after the decimal places) and one integer *P* ($0 \le P \le 100$). These two numbers denote a file whose size is *S* megabyte and which has been downloaded exactly *P*% already. Also note that although theoretically it is not possible that the size of a file or size of its remaining part is a fraction when expressed in bytes, for simplicity please assume that such thing is possible in this problem. The last test case is followed by $T=n=B=0$, which should not be processed.

Output
For each case, print the case number and the total time required to download all the files, expressed in hours and rounded to 2 digits after the decimal point. Print a blank line after the output of each test case.

Sample Input	Output for the Sample Input
6 3 90 100.00 90 40.40 70 60.30 70 40.40 80 40.40 85 40.40 88 1 1 56 12.34 100 0 0 0	Case 1: 0.66 Case 2: 0.00

Explanation
In the first sample, there are 6 files and the download manager can download 3 files simultaneously. The size of the smallest file is 40.40 Megabyte but there are four such files (2nd, 4th, 5th and 6th files). So the download manager chooses the 6th, 5th and 4th files for download as they have less bytes remaining. All these files get equal bandwidth (30.00 Megabyte/Sec). Of these three files the 8th file is finished first. So instantaneously the 2nd file starts downloading. Then, 5th file is finished. So the next larger file (3rd file) starts downloading. This process goes on until all files are downloaded.