



## 3387 - Job Scheduling by Open Bidding

North America - Pacific Northwest - 2005/2006

Your team is setting up a computing resource devoted to batch processing of compute-bound jobs. In addition, you have decided to use static scheduling for each period of time. Naturally, you wish to maximize the income for each set of jobs run, and you have been given the responsibility of finding an optimal mix of jobs for each set of candidate jobs. The jobs are submitted by an open bid process: clients will specify the amount of processor time they wish to reserve and the dollar amount that they wish to pay. If a job finishes early, the client will still pay the full amount, and if a job exceeds the requested time, it will be terminated and (of course) the client will still pay the full amount. For purposes of scheduling, your team assumes that each job will in fact use its entire scheduled time slot. In the interests of good customer relations, though, you are not to include a bid in the schedule if there is not sufficient time available to satisfy it - we're not going to over-book like the airlines do, and then hope someone doesn't use the full allotment!

### Input

The input file begins with a line containing a single integer (no white space) specifying the number of problem sets in the file.

Each problem set consists of  $(n + 2)$  lines (no white space except as specified):

- a single integer  $n$  ( $n \leq 500$ ) specifying the number of candidate jobs to be scheduled
- $n$  lines giving the bid as an integer specifying the number of seconds followed by a single space and then a dollar amount given in decimal form (always showing two digits to the right of the decimal point)
- a single integer  $t$  ( $t \leq 2000$ ) specifying the amount of time to be scheduled with these jobs

### Output

Each problem set will be numbered (beginning at one) and will generate a single line:

```
Problem < k > : < t > seconds scheduled for $abc.de
```

where  $< k >$  is replaced by the problem set number,  $< t >$  is replaced with the total time actually scheduled (possibly not the full input time), and  $\$abc.de$  is replaced by the dollar amount, given always with the leading currency symbol and with two digits to the right of the decimal point. There will be no blank lines, and the final line will end with the new-line character.

### Sample Input

```
1
10
19 0.78
12 0.31
17 0.77
```

22 0.77  
8 0.56  
10 0.33  
17 0.35  
24 0.12  
22 0.70  
5 0.52  
120

## Sample Output

Problem 1: 120 seconds scheduled for \\$.78

---

Pacific Northwest 2005-2006