Escalonador round-robin

Robin é um jovem apaixonado por programação e sistemas operacionais, em seus estudos mais recentes está aprendendo junto com seu amigo Tanenbaum o funcionamento dos escalonadores dos sistemas operacionais

Para ter certeza que entendeu o problema corretamente, Robin decidiu implementar por conta própria um escalonador para imprimir a ordem de execução de uma sequência de processos aleatórios.

Nesta tarefa, você deve implementar o algoritmo de escalonamento preemptivo round-robin, que recebe como entrada o número de processos, uma janela de tempo e o tempo de execução de cada processo e ao final imprime quando cada processo termina a execução.

Entrada

A primeira linha da entrada é um número inteiro N, entre 1 e 100, indicando o número de processos que serão escalonados. A segunda linha contém a janela de tempo (1 <= T <= 1000) em MILISSEGUNDOS. As próximas N linhas da entrada possuem um identificador único (pid) e o tempo total de execução (t) em SEGUNDOS que esse processo precisa para executar (1 <= pid <= 200, 1 <= t <= 60000).

Saída

A saída contém N linhas, onde os processos são impressos na ordem que terminaram a execução. Considere que todos os processos iniciam a execução no tempo 0. Para cada um dos processos, também é impresso entre parênteses o tempo total quando o processo terminou a execução, ou seja, o turnaround de cada processo.

Exemplos

Exemplo de entrada

```
1
500
1 1
```

Saída para o exemplo de entrada

```
1 (1000)
```

Exemplo de entrada

```
2
500
1 2
2 1
```

Saída para o exemplo de entrada

```
2 (2000)
1 (3000)
```

Exemplo de entrada

```
3
500
23 6
186 2
59 2
```

Saída para o exemplo de entrada

```
186 (5500)
59 (6000)
23 (10000)
```

Exemplo de entrada

```
10
1
1 10000
2 1
3 1
4 1
5 1
6 1
7 1
8 1
9 1
10 60000
```

Saída para o exemplo de entrada

PS: Essa saída deve ser produzida em menos de 2 segundos

```
2 (9992)
3 (9993)
4 (9994)
5 (9995)
6 (9996)
7 (9997)
8 (9998)
9 (9999)
1 (20007999)
10 (70008000)
```

Linguagens de programação

Para performance, deve ser feita a implementação em C (submeter arquivo com extensão .c) ou C++ (extensão .cpp).

Author: Daniel Sundfeld <daniel.sundfeld@unb.br>