

Montanha russa (use queue)

O parque de diversões da Labilândia sempre inova em suas atrações e o mais novo sucesso é a montanha russa do loop quebrado, a mais alta e mais rápida do mundo.

O sucesso é tamanho que filas intermináveis se formam todos os dias. As pessoas não se contentam em andar apenas uma vez, assim que saem do brinquedo já entram novamente na fila.

A entrada na montanha russa custa 'X' dinheiros, cada volta dura apenas 1 minuto e apenas uma pessoa anda por vez. Para

otimizar o funcionamento, o parque te contratou para desenvolver um algoritmo de controle da fila. Decorridos 'M' minutos após a abertura do parque, o seu programa deverá informar o estado atual da fila, dizendo o identificador das pessoas que ainda estão na fila e quanto dinheiro aquelas pessoas ainda possuem. Lembre-se que uma pessoa só entra na fila se tiver dinheiro suficiente para pagar a entrada.



Entrada

Seu programa receberá na primeira linha um inteiro 'N' ($1 \leq N \leq 10^6$) representando a quantidade de pessoas inicialmente na fila da montanha russa na abertura do parque, um inteiro 'M' ($1 \leq M \leq 900$), o tempo decorrido desde a abertura do parque, e um inteiro 'X' ($1 \leq X \leq 100$) representando o valor do ingresso para andar na montanha russa.

Cada uma das 'N' linhas seguintes terá primeiro um inteiro 'Y' ($1 \leq Y \leq 10^6$), representando o identificador da pessoa 'Yi', seguido de um inteiro 'D' ($1 \leq D \leq 1000$), representando a quantidade de dinheiro que a pessoa 'Yi' possui.

Saída

Seu programa deve apresentar o estado atual da fila, decorridos 'M' minutos desde a abertura do parque. Imprima o identificador da pessoa e a quantidade de dinheiro ainda em sua posse, imprima uma pessoa por linha. Considere dinheiro gasto apenas se a pessoa andou no brinquedo. Caso a fila esteja vazia imprima "SEM FILA", tudo maiúsculo e sem aspas.

Obs.:

Sugerimos o uso da seguinte struct:

```
struct pessoa{
    int id;
    int dinheiro;
};
```

Exemplos:

Entrada	Saída
3 2 5 1 10 2 20 3 30	3 30 1 5 2 15
4 3 10 1 10 3 10 2 10 4 5	SEM FILA