Na empresa ABC, temos uma relação de tarefas a serem desenvolvidas, essas tarefas estão definidas em uma lista na forma: [(a,b)], tal que a é o nome da tarefa e b é a prioridade com a qual ela deve ser desenvolvida. Em um exemplo hipotético, teríamos: [("Tarefa A",1),("Tarefa X",2),("Tarefa G",7),("Tarefa C",2),("Tarefa K",4)]

Note que duas tarefas podem ter mesma prioridade, e neste caso a ordem de execução é definida pela sua posição na lista.

No decorrer do tempo, essa lista de tarefas vai sofrendo alterações, com inclusões e exclusões de novas tarefas, e a prioridade das tarefas pode ter intervalos. Isso ocorre porque novas tarefas são inclusas no final da lista, e podem ter prioridade maiores ou menores daquelas que já temos na lista.

O valor das prioridades deve estabelecer a ordem com que as tarefas que estão na lista devem ser executadas, portanto esses pesos podem ser alterados desde que não alterem a ordem que as tarefas devem ser executadas. Por exemplo a lista anterior pode ser reescrita alterando os pesos sem que a ordem seja alterada da seguinte forma:
[("Tarefa A",1),("Tarefa X",2),("Tarefa G",4),("Tarefa C",2),("Tarefa K",3)]

Desenvolva um programa que, dada uma lista de tarefas de tipo [([Char],Int)], retorne a lista de tarefas com a prioridade atualizada, sem que a ordem seja afetada.

Para isso, algumas etapas são necessárias:

1. Desenvolva a função geraPosicaoDosItens, que recebe uma lista de tarefas e prioridades, a posição inicial que é 1, e retorna uma lista de tarefas com tipo: [(a,b,c)], onde a é nome da tarefa, b é a prioridade, e c a ordem do elemento na lista.
*geraPosicaoDosItens :: [([Char],Int)] -> Int -> [([Char],Int,Int)]*
Ex: geraPosicaoDosItens [("Tarefa A",1),("Tarefa X",2),("Tarefa G",7),("Tarefa C",2),("Tarefa K",4)] 1
>> [("Tarefa A",1,1),("Tarefa X",2,2),("Tarefa G",7,3),("Tarefa C",2,4),("Tarefa K",4,5)]

2. Desenvolva a função ordenaListaPelaPrioridade, que recebe uma lista de tipo [(a,b,c)]- conf. item 1, e retorna uma lista de mesmo tipo ordenada pela prioridade.
*ordenaListaPelaPrioridade :: [([Char],Int,Int)] -> [([Char],Int,Int)]*
Ex: ordenaListaPelaPrioridade [("Tarefa A",1,1),("Tarefa X",2,2),("Tarefa G",7,3),("Tarefa C",2,4),("Tarefa K",4,5)]
>> [("Tarefa A",1,1),("Tarefa X",2,2),("Tarefa C",2,4),("Tarefa K",4,5),("Tarefa G",7,3)]

3. Desenvolva a função atualizaPrioridade, que recebe uma lista de tipo [(a,b,c)]- conf. item 1 ordenada pela prioridade, e a prioridade inicial que é 1, e retorna uma lista de mesmo tipo com as prioridades atualizadas, iniciando por 1.
*atualizaPrioridade :: [([Char],Int,Int)] -> Int -> [([Char],Int,Int)]*
Ex: atualizaPrioridade [("Tarefa A",1,1),("Tarefa X",2,2),("Tarefa C",2,4),("Tarefa K",4,5),("Tarefa G",7,3)] 1
>> [("Tarefa A",1,1),("Tarefa X",2,2),("Tarefa C",2,4),("Tarefa K",3,5),("Tarefa G",4,3)]

4. Desenvolva a função ordenaListaPelaOrdem, que recebe uma lista de tipo [(a,b,c)]- conf. item 1 com a prioridade atualizada, e retorna uma lista de mesmo tipo ordenada pela sua posição original (terceiro elemento da tupla).
*ordenaListaPelaOrdem :: [([Char],Int,Int)] -> [([Char],Int,Int)]*
Ex: ordenaListaPelaOrdem [("Tarefa A",1,1),("Tarefa X",2,2),("Tarefa C",2,4),("Tarefa K",3,5),("Tarefa G",4,3)]
>> [("Tarefa A",1,1),("Tarefa X",2,2),("Tarefa G",4,3),("Tarefa C",2,4),("Tarefa K",3,5)

O teste final, será feito pela função imprimeListaAtualizada, cuja implementação está abaixo. Ela recebe uma lista de tarefas e prioridades retorna uma lista de mesmo tipo, porém com a prioridade atualizada. O retorno deve ser adaptado para o tipo indicado.

```
-- imprimeListaAtualizada :: [([Char],Int)]-> [([Char],Int)]
imprimeListaAtualizada xs = ordenaListaPelaOrdem (atualizaPrioridade (ordenaListaPelaPrioridade (geraPosicaoDosItens xs 1)) 1)
```

Esse trabalho é individual, e deve ser entregue um arquivo .hs via moodle.