

MC346 - PROJETO HASKELL
Lei de Benfort

Escreva um programa em haskell que conta o número de vezes que cada dígito (1 a 9) aparece como o primeiro dígito em um texto qualquer. O texto pode conter palavras, pontuação e números. Os números so podem estar em 4 formatos:

- uma sequencia de digitos (por exemplo 5457)
- uma sequencia de digitos com um ponto decimal (545.7)
- uma sequencia de digitos (com ou sem ponto) iniciada por “-” (-54.57)
- uma sequencia (com ou sem ponto decimal) iniciada por “R\$” (R\$54.57)

Em todos os casos acima, só estamos interessados no primeiro dígito (o 5).

Além disso, vamos assumir que os números no texto (nos 4 formatos acima) estão separados do resto do texto por brancos (ou inicio/fim de linha). Assim não há no texto coisas como:

```
0 kilo de tomate esta a R$8.50!!  
0 preco do tomate e R$8.50/kg
```

A saída deve ser no seguinte formato:

```
1 : n1  
2 : n2  
3 : n3  
4 : n4  
5 : n5  
6 : n6  
7 : n7  
8 : n8  
9 : n9
```

Onde n_1 é o número de vezes que o 1 foi o primeiro digito. Os números n_i estao alinhados a esquerda, por exemplo:

```
1 : 176  
2 : 4  
3 : 0  
...
```

que indica que o 1 é o primeiro digito 176 vezes, o 2 em 4 vezer e o 3 nenhuma. Note que há uma linha em branco ao final da saída

O susy rodara seu programa como:

```
ghc ex2.hs  
ex2 < arqx.in
```

Se voce não conseguir tratar os casos onde os números podem começar com “-” ou “R\$” (que são o 3o caso de teste aberto e o 3o caso de teste fechado) o programa valerá no máximo 6.