

```

| otherwise = y
  where
    x1 = idade (pessoa x)
    x2 = idade (pessoa y)

```

Execução:

```

Main> maior_idade 6
6.0
Main> maior_idade 5 -- a maior idade até o 5º registro
2.0 -- é o registro 2.

```

4.4 Exercícios Propostos

1. Defina uma função que retorne uma tupla-3 (*tripla*) contendo o caractere fornecido com entrada, o mesmo caractere em letras minúsculas ou maiúsculas, e o seu número da tabela ASCII. Exemplo:

converte b (b,B,98)

2. Seja o cadastro de pessoas dado pela função a seguir:

```

pessoa rg | rg == 1 = ("João Silva", 12, 'm')
          | rg == 2 = ("Jonas Souza", 51, 'm')
          | .....
          | rg == 321 = ("Jocileide Strauss", 21, 'f')
          | otherwise = ("Não há ninguém mais", 9999, 'x')

```

Construa funções que retornem os seguintes dados:

- (a) O nome da pessoa de menor idade até um determinado registro.
- (b) A idade média de todas as pessoas até um dado registro.
- (c) O número de pessoas do sexo masculino.
- (d) O número do registro da pessoa de maior idade.

3. Construa uma função em que, dado um caractere qualquer, retorne uma tupla-3 com o caractere dado, o caractere dado na forma maiúscula/minúscula (o contrário do original) e o número ASCII do original. Exemplos:

```

Main> analisaLetra 'h'
('h', 'H', 104)
Main> analisaLetra 'H'
('H', 'h', 72)

```

4. Construa uma função em Haskell que recebe 4 inteiros e devolve uma tupla-4 com os quatro valores originais, só que ordenados. Exemplo:

```

Main> ordena 3 5 1 (-3)
(-3, 1, 3, 5)

```

5. Dadas duas datas (d_1, m_1, a_1) e (d_2, m_2, a_2) , tal que $data_1 \leq data_2$, construa uma função que retorne quantos dias existem entre estas duas datas, onde d_i define o dia do mês m_j no ano a_k .

- ✓6. Crie uma função que receba os coeficientes de uma equação do segundo grau $ax^2 + bx + c = 0$ na forma (a,b,c) e retorne as raízes desta equação. Trate o caso de raízes imaginárias, indicando um erro.

Exemplo:

```
Main> equacao (1,(-5),6)
(2,3)
```

- ✓7. Construa uma função que, dados três valores, verifique se os mesmos podem ser os lados de um triângulo. Se for possível formar o triângulo, retorne uma tupla-2 com o tipo do triângulo formado (com relação às arestas) e o perímetro do mesmo. Exemplo:

```
Main> triangulo (7,7,11)
("Isóceles",25)
```

- ✓8. Apresentada uma base de dados de 10 professores:

```
base :: Int -> (Int, String, String, Char)
```

```
base x
```

x == 0	= (1793, "Pedro Paulo",	"MESTRE", 'M')
x == 1	= (1797, "Joana Silva Alencar",	"MESTRE", 'M')
x == 2	= (1534, "João De Medeiros",	"DOUTOR", 'F')
x == 3	= (1267, "Cláudio César de Sá",	"DOUTOR", 'M')
x == 4	= (1737, "Paula de Medeiros",	"MESTRE", 'F')
x == 5	= (1888, "Rita de Matos",	"MESTRE", 'F')
x == 9	= (1698, "Tereza Cristina Andrade",	"MESTRE", 'F')
x == 10	= (0, "",	"", 'O')

Construa funções que retornem:

- O número de doutores na base.
- O número de mulheres.
- O número de mestres do sexo masculino.
- O nome do professor mais antigo (número de menor matrícula).