

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

```
import Char
type Pessoa = (String, Int, Float, Char)

pess :: Int->Pessoa
pess x
  |x==1 = ("Rosa", 27, 1.66, 'F')
  |x==2 = ("João", 26, 1.85, 'M')
  |x==3 = ("Maria", 67, 1.55, 'F')
  |x==4 = ("Jose", 48, 1.78, 'M')
  |x==5 = ("Paulo", 24, 1.93, 'M')
  |x==6 = ("Clara", 38, 1.70, 'F')
  |x==7 = ("Bob", 12, 1.45, 'M')
  |x==8 = ("Rosana", 31, 1.58, 'F')
  |x==9 = ("Daniel", 75, 1.74, 'M')
  |x==10 = ("Jocileide", 21, 1.69, 'F')
  |otherwise = ("Acabou!", 0, 0.0, 'x')
```

Construa funções que retornem os seguintes dados:

- O número do registro da pessoa de maior idade.

```
Main> maiorIdade 6
```

- A idade média de todas as pessoas.

```
Main> idadeMedia 4
```

- O número de pessoas do sexo masculino com idade superior a 25 anos.

```
Main> mascSup25 8
```

2. Construa uma função que, dado 3 valores, verifique se os mesmos podem ser os lados de um triângulo. Se for possível formar o triângulo, retorne um tupla-2 com o tipo de triângulo formado (com relação às arestas) e o perímetro do mesmo.

```
Main> triangulo (7,7,11)
("Isosceles",25.0)
Main> triangulo (7,7,7)
("Equilatero",21.0)
Main>
```

3. Faça uma função que calcule a soma de todos os números ímpares de 1 à N, utilizando recursão de cauda.

```
Main> somaP 10
30
Main>
```

4. Faça uma função que calcula a série:

$$Y = \frac{1}{x} + \frac{x}{2} + \frac{3}{x} + \frac{x}{4} + \dots$$

$$y = (1/x) + (x/2) + (3/x) + (x/4) + \dots$$

A chamada da função será um tupla-2, onde o primeiro campo será o valor de x e o segundo campo será a quantidade de termos da série. Obs. A função deverá ter recursão de cauda.

```
Main> serie (1,100)
2502.25
```

```
Main> serie (2,100)
1254.499
Main>
```