

Nome: GUILHERME LUCAS PEREIRA BERNARDO

## LISTA DE EXERCÍCIO

1. Leia 3 valores de ponto flutuante A, B e C e ordene-os em ordem decrescente, de modo que o lado A representa o maior dos 3 lados. A seguir, determine o tipo de triângulo que estes três lados formam, com base nos seguintes casos, sempre escrevendo uma mensagem adequada:
  - se  $A \geq B+C$ , apresente a mensagem: **NAO FORMA TRIANGULO**
  - se  $A^2 = B^2 + C^2$ , apresente a mensagem: **TRIANGULO RETANGULO**
  - se  $A^2 > B^2 + C^2$ , apresente a mensagem: **TRIANGULO OBTUSANGULO**
  - se  $A^2 < B^2 + C^2$ , apresente a mensagem: **TRIANGULO ACUTANGULO**
  - se os três lados forem iguais, apresente a mensagem: **TRIANGULO EQUILATERO**
  - se apenas dois dos lados forem iguais, apresente a mensagem: **TRIANGULO ISOSCELES**

Exemplos de Entrada	Exemplos de Saída
7.0 5.0 7.0	TRIANGULO ACUTANGULO TRIANGULO ISOSCELES
6.0 6.0 10.0	TRIANGULO OBTUSANGULO TRIANGULO ISOSCELES
6.0 6.0 6.0	TRIANGULO ACUTANGULO TRIANGULO EQUILATERO
5.0 7.0 2.0	NAO FORMA TRIANGULO
6.0 8.0 10.0	TRIANGULO RETANGULO

R:

```
import Text.Printf (printf)
import Data.List (reverse, sort)

isTriangle::[Float]->Bool
isTriangle x = x!!0 < (x!!1 + x!!2)

tipoTrianguloUm::[Float] -> String
tipoTrianguloUm x = let ladoa = x!!0
                    ladob = x!!1
                    ladoc = x!!2
                    in if ladoa * ladoa == ladob * ladob + ladoc * ladoc
then "TRIANGULO RETANGULO"
    else if ladoa * ladoa > ladob * ladob + ladoc * ladoc
then "TRIANGULO OBTUSANGULO"
    else "TRIANGULO ACUTANGULO"

tipoTrianguloDois::[Float]->String
tipoTrianguloDois x = let ladoa = x !!0
                    ladob = x !!1
                    ladoc = x !!2
```

```

                                in if ladoa == ladob && ladob == ladoc then "TRIANGULO
EQUILATERO"
                                else if ladoa /= ladob && ladob /= ladoc && ladoa /=
ladoc then ""
                                else "TRIANGULO ISOCELES"

ordDecrescente :: String->[Float]
ordDecrescente x = let ladoa = read (words x !!0)::Float
                    ladob = read (words x !!1)::Float
                    ladoc = read (words x !!2)::Float
                    listSides = [ladoa, ladob, ladoc]
                    in reverse (sort listSides)

main :: IO()
main = do x <- getLine :: IO String
        let listaOrd = ordDecrescente x
        if not (isTriangle listaOrd) then printf "NAO FORMA TRIANGULO"
        else do
            print $ tipoTrianguloUm listaOrd
            print $ tipoTrianguloDois listaOrd

```

2. Escreva um programa para ler as coordenadas (X,Y) de uma quantidade indeterminada de pontos no sistema cartesiano. Para cada ponto escrever o quadrante a que ele pertence. O algoritmo será encerrado quando pelo menos uma de duas coordenadas for zero (nesta situação sem escrever mensagem alguma).

Exemplo de Entrada	Exemplo de Saída
2 2	primeiro
3 -2	quarto
-8 -1	terceiro
-7 1	segundo
0 2	

R:

```

import Text.Printf
--decidequad :: Integer

main :: IO ()
main = do vs <- getLine :: IO String
        let [a, b] = map (read :: String -> Int) (words vs)
        if (a>0 && b>0)
            then putStrLn "primeiro quadrante"
        else if (a<0 && b>0)
            then putStrLn "segundo quadrante"
        else if (a<0 && b<0)
            then putStrLn "terceiro quadrante"
        else if (a>0 && b<0)
            then putStrLn "quarto quadrante"
        else if (a==0 || b==0)
            then putStrLn ""
        else putStrLn ""

```

3. Escreva um programa que leia um valor inteiro N. Este N é a quantidade de linhas de saída que serão apresentadas na execução do programa como ilustrado no exemplo abaixo.

Exemplo de Entrada	Exemplo de Saída
7	1 2 3 PUM 5 6 7 PUM 9 10 11 PUM 13 14 15 PUM 17 18 19 PUM 21 22 23 PUM 25 26 27 PUM

R:

```
func :: Int -> Int -> IO()
soma c = func c 0
func c s =
    if (c > 0) then do
        print (s + 1, s + 2, s + 3, "PUM")
        func (c-1) (s+4)
    else do
        return ()
```

4. Na matemática, um Número Primo é aquele que pode ser dividido somente por 1 (um) e por ele mesmo. Por exemplo, o número 7 é primo, pois pode ser dividido apenas pelo número 1 e pelo número 7. Escreva uma função que recebe um número com entrada e informa se ele é primo ou não. Esse algoritmo deve verificar até os primeiros 100 números. Se um valor maior que 100 for dado como entrada exiba a seguinte mensagem: "Atribua um valor entre 0 e 100"

R:

```
processa a = [ x | x <- [ 1..a ], mod a x == 0 ]

prime a = if ( a < 99 ) then
    if ( processa a == [ 1, a ] ) then
        print( "True" )
    else print( "False" )
    else print( "Atribua um valor entre 0 e 100" )
```

5. Neste problema, deve-se ler o código de uma peça 1, o número de peças 1, o valor unitário de cada peça 1, o código de uma peça 2, o número de peças 2 e o valor unitário de cada peça 2. Após, calcule e mostre o valor a ser pago.

8	51 nao eh primo
51	7 eh primo

R:

```
import Text.Printf

main :: IO ()
main = do linha1 <- getLine :: IO String
        linha2 <- getLine :: IO String
        let [ codigo1, numero1, valor1 ] = map ( read :: String -> Int ) $
words linha1
        let [ codigo2, numero2, valor2 ] = map ( read :: String -> Int ) $
words linha2
        printf "Total a pagar: %d\n" $ numero1 * valor1 + numero2 * valor2
```

6. Uma empresa decidiu dar a seus funcionários um abono de salário, baseando-se nos pontos obtidos durante o mês, de acordo com a tabela:

<i>Pontos Obtidos</i>	<i>Prêmio em R\$</i>
<i>1 a 10</i>	<i>100,00</i>
<i>11 a 20</i>	<i>200,00</i>
<i>21 a 30</i>	<i>300,00</i>
<i>31 a 40</i>	<i>400,00</i>
<i>A partir de 41</i>	<i>500</i>

Crie um função no haskell que recebe como entrada o salário do funcionário e os pontos obtidos e que retorne a quantidade total que ele vai receber (salário mais prêmio).

Exemplos de Entrada	Exemplos de Saída
7 14 106	106 eh o maior
217 14 6	217 eh o maior

R:

```
premio salario pontosObtidos | ( pontosObtidos > 0 && pontosObtidos < 11 ) =
salario + 100
    | ( pontosObtidos > 10 && pontosObtidos < 21 ) = salario + 200
    | ( pontosObtidos > 20 && pontosObtidos < 31 ) = salario + 300
    | ( pontosObtidos > 30 && pontosObtidos < 41 ) = salario + 400
    | ( pontosObtidos > 40 ) = salario + 500
```