

LISTA DE EXERCÍCIO

1. A fórmula para calcular a área de uma circunferência é: $area = \pi \cdot raio^2$. Considerando para este problema que $\pi = 3.14159$:

Efetue o cálculo da área, elevando o valor de raio ao quadrado e multiplicando por π .

R:

```
-- module Main where
import Text.Printf ( printf )
area :: Double -> Double
area r = 3.14159*r**2
main :: IO ()
main = do r <- readLn :: IO Double
        printf "A=%.4f\n" $ area r
```

2. Leia dois valores inteiros. A seguir, calcule o produto entre estes dois valores e atribua esta operação à variável PROD. A seguir mostre a variável PROD com mensagem correspondente.

Entrada

O arquivo de entrada contém 2 valores inteiros.

Saída

Imprima a mensagem "PROD" e a variável PROD conforme exemplo abaixo, com um espaço em branco antes e depois da igualdade.

Exemplos de Entrada	Exemplos de Saída
3 9	PROD = 27
-30 10	PROD = -300
0 9	PROD = 0

R:

```
import Text.Printf

main :: IO ()
main = do a <- readLn :: IO Int
        b <- readLn :: IO Int
        printf "PROD = %d\n" $ a*b
```

3. Leia 3 valores, no caso, variáveis A, B e C, que são as três notas de um aluno. A seguir, calcule a média do aluno, sabendo que a nota A tem peso 2, a nota B tem peso 3 e a nota C tem peso 5. Considere que cada nota pode ir de 0 até 10.0, sempre com uma casa decimal.

Entrada

O arquivo de entrada contém 3 valores com uma casa decimal, de dupla precisão (double).

Imprima a mensagem "MEDIA" e a média do aluno conforme exemplo abaixo, com 1 dígito após o ponto decimal e com um espaço em branco antes e depois da igualdade

Exemplos de Entrada	Exemplos de Saída
5.0 6.0 7.0	MEDIA = 6.3
5.0 10.0 10.0	MEDIA = 9.0
10.0 10.0 5.0	MEDIA = 7.5

R:

```
--module Main where
import Text.Printf

main :: IO ()
main = do putStrLn "Digite um numero: "
          s1 <- readLn :: IO Double
          putStrLn "Digite um segundo numero: "
          s2 <- readLn :: IO Double
          putStrLn "Digite outro numero: "
          s3 <- readLn :: IO Double
          printf "MEDIA DOS NUMEROS= %f\n" (((s1*2) + (s2*3) +
(s3*5))/10)
```

- Escreva um programa que leia o número de um funcionário, seu número de horas trabalhadas, o valor que recebe por hora e calcula o salário desse funcionário. A seguir, mostre o número e o salário do funcionário, com duas casas decimais.

Entrada

O arquivo de entrada contém 2 números inteiros e 1 número com duas casas decimais, representando o número, quantidade de horas trabalhadas e o valor que o funcionário recebe por hora trabalhada, respectivamente.

Saída

Imprima o número e o salário do funcionário, conforme exemplo fornecido, com um espaço em branco antes e depois da igualdade. No caso do salário, também deve haver um espaço em branco após o \$.

Exemplos de Entrada	Exemplos de Saída
25 100 5.50	NUMBER = 25 SALARY = U\$ 550.00
1 200 20.50	NUMBER = 1 SALARY = U\$ 4100.00
6 145 15.55	NUMBER = 6 SALARY = U\$ 2254.75

R:

```
--module Main where
import Text.Printf ( printf )

main :: IO ()
main = do s1 <- readLn ::IO Int
          s2 <- readLn ::IO Double
          s3 <- readLn ::IO Double
          printf "NUMERO = %d\n" s1
          printf "SALARIO = U$ %.1f\n" $ s2*s3
```

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.

Entrada

O arquivo de entrada contém duas linhas de dados. Em cada linha haverá 3 valores, respectivamente dois inteiros e um valor com 2 casas decimais.

Saída

A saída deverá ser uma mensagem conforme o exemplo fornecido abaixo, lembrando de deixar um espaço após os dois pontos e um espaço após o "R\$". O valor deverá ser apresentado com 2 casas após o ponto.

Exemplos de Entrada	Exemplos de Saída
12 1 5.30 16 2 5.10	VALOR A PAGAR: R\$ 15.50
13 2 15.30 161 4 5.20	VALOR A PAGAR: R\$ 51.40
1 1 15.10 2 1 15.10	VALOR A PAGAR: R\$ 30.20

R:

```
import Text.Printf
```

```

total :: [String] -> Double
total w = let n = read (w !! 1) :: Double
           p = read (w !! 2) :: Double
           in n * p

main :: IO ()
main = do vs1 <- getLine :: IO String
          vs2 <- getLine :: IO String
          let t1 = total $ words vs1
          let t2 = total $ words vs2
          printf "VALOR A PAGAR: R$ %.2f\n" $ t1 + t2

```

6. Escreva um programa que leia três valores com ponto flutuante de dupla precisão: A, B e C. Em seguida, calcule e mostre:
- a) a área do triângulo retângulo que tem A por base e C por altura.
 - b) a área do círculo de raio C. ($\pi = 3.14159$)
 - c) a área do trapézio que tem A e B por bases e C por altura.
 - d) a área do quadrado que tem lado B.
 - e) a área do retângulo que tem lados A e B.

R:

```

module Main where

main :: IO ()
main = do ia <- readLn :: IO Double
          ib <- readLn :: IO Double
          ic <- readLn :: IO Double
          putStr "Area do triangulo: "
          print (ia * ic / 2)
          putStr "Area do circulo de raio C: "
          print (3.14159 * ic * ic)
          putStr "Area do trapezio: "
          print ((ia + ib) * (ic / 2))
          putStr "Area do quadrado: "
          print (ib * ib)
          putStr "Area do retangulo: "
          print (ia * ib)

```

7. Faça um programa que leia três valores e apresente o maior dos três valores lidos seguido da mensagem “é o maior”. Utilize a fórmula:

$$MaiorAB = \frac{(a+b+abs(a-b))}{2}$$

Obs.: a fórmula apenas calcula o maior entre os dois primeiros (a e b). Um segundo passo, portanto é necessário para chegar no resultado esperado.

Entrada

O arquivo de entrada contém três valores inteiros.

Saída

Imprima o maior dos três valores seguido por um espaço e a mensagem "eh o maior".

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

R:

```
import Text.Printf

maiorAB :: Int -> Int -> Int
maiorAB a b = (a + b + abs (a - b)) `div` 2

main :: IO ()
main = do vs <- getLine :: IO String
        let [a, b, c] = map (read :: String -> Int) (words vs)
        print $ maiorAB a (maiorAB b c)
```

8. Leia os quatro valores correspondentes aos eixos x e y de dois pontos quaisquer no plano, p1(x1, y1) e p2(x2, y2) e calcule a distância entre eles, segundo a fórmula:

$$\text{Distância} = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$

Entrada

O arquivo de entrada contém duas linhas de dados. A primeira linha contém dois valores de ponto flutuante: x1 y1 e a segunda linha contém dois valores de ponto flutuante x2 y2.

Saída

Calcule e imprima o valor da distância segundo a fórmula fornecida.

Exemplo de Entrada	Exemplo de Saída
1.0 7.0 5.0 9.0	4.4721
-2.5 0.4 12.1 7.3	16.1484
2.5 -0.4 -12.2 7.0	16.4575

R:

```
import Text.Printf ( printf )
```

```

main :: IO ()
main = do ix1 <- readLn :: IO Double
         iy1 <- readLn :: IO Double
         ix2 <- readLn :: IO Double
         iy2 <- readLn :: IO Double
         printf "%.2f\n" $ sqrt
((ix2-ix1)*(ix2-ix1)+(iy2-iy1)*(iy2-iy1))

```

9. Leia um valor inteiro N. Este valor será a quantidade de valores que serão lidos em seguida. Para cada valor lido, mostre uma mensagem em inglês dizendo se este valor lido é par (EVEN), ímpar (ODD), positivo (POSITIVE) ou negativo (NEGATIVE). No caso do valor ser igual a zero (0), embora a descrição correta seja (EVEN NULL), pois por definição zero é par, seu programa deverá imprimir apenas NULL.

Exemplo de Entrada	Exemplo de Saída
4	ODD NEGATIVE
-5	NULL
0	ODD POSITIVE
3	EVEN NEGATIVE
-4	

R:

```

import Text.Printf

-- rem - retorna a divisão inteira dos argumentos
-- abs - retorna um valor absoluto de um número

isPositive :: Int -> String
isPositive n = if (n >= 0) then "POSITIVE" else "NEGATIVE"

isOdd :: Int -> String
isOdd n = if (rem (abs(n)) 2 == 1) then "EVEN " else "ODD "

myodd :: Integer -> Bool
myodd n = rem (abs(n)) 2 == 1

main :: IO ()
main = do a <- readLn :: IO Int
         if (a /= 0) then print((isOdd a) ++ (isPositive a)) else
print("NULL")
         b <- readLn :: IO Int
         if (b /= 0) then print((isOdd b) ++ (isPositive b)) else
print("NULL")
         c <- readLn :: IO Int
         if (c /= 0) then print((isOdd c) ++ (isPositive c)) else
print("NULL")

```

```
    d <- readLn :: IO Int
    if (d /= 0) then print((isOdd d) ++ (isPositive d)) else
print("NULL")
```

10. Leia 6 valores. Em seguida, mostre quantos destes valores digitados foram positivos. Na próxima linha, deve-se mostrar a média de todos os valores positivos digitados.

R:

```
module Main where

import Text.Printf

letPositive :: Double -> Double

letPositive n = if n >= 0 then n else 0

sumArr :: [Double] -> Double

sumArr [] = 0

sumArr (a:x) = a + sumArr x

main :: IO ()

main = do a <- readLn :: IO Double
         b <- readLn :: IO Double
         c <- readLn :: IO Double
         d <- readLn :: IO Double
         e <- readLn :: IO Double
         f <- readLn :: IO Double

         let allPositives = map letPositive [a, b, c, d, e, f]

         print(sumArr allPositives)

         print $ sumArr allPositives / 6
```