

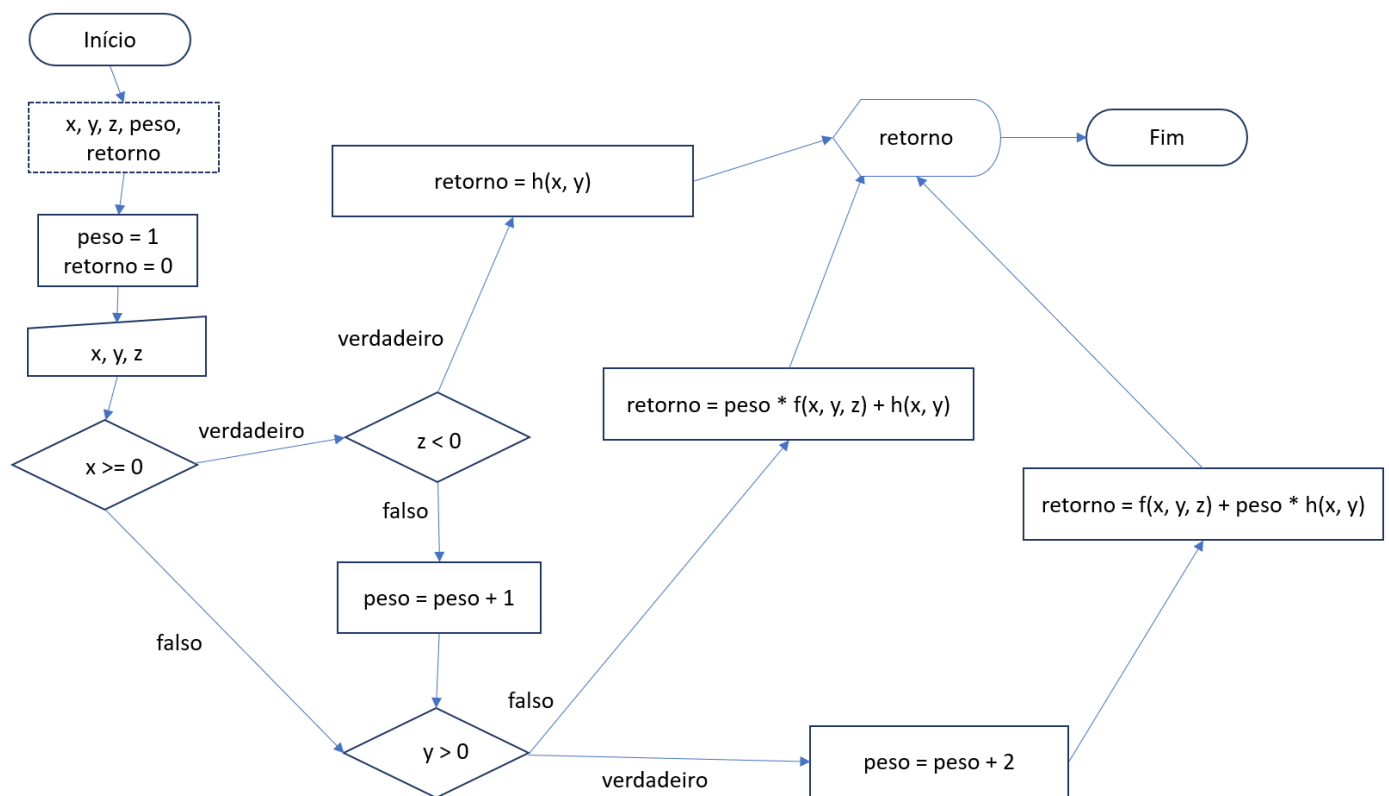
Nome: \_\_\_\_\_ Matrícula: \_\_\_\_\_ IP: \_\_\_\_\_

Você recebeu o enunciado do Teste 2 da G1 que deve ser **resolvido em sala de aula e enviado pela plataforma EAD**.

O teste contém três questões. Ele é individual, tem a duração de **50 minutos** e todas as atividades relacionadas à solução do trabalho proposto devem ser realizadas, respeitando-se o código de ética do CTC disponível na plataforma EAD.

Para cada questão do teste deverá ser criado um script correspondente que deverá ser salvo com o nome “INF1514\_TEAG1\_QX\_MATRICULA.R”, substituindo o texto “MATRICULA” pelo número da sua matrícula e X pelo número da questão. Cada script deverá conter todas as implementações realizadas para a correspondente questão, incluindo os testes, sendo que a criação e organização dos scripts faz parte da resolução do trabalho.

Considere o seguinte fluxograma criado por um analista para processar dados coletados de um experimento.



Sabendo que  $f(x, y, z) = x^3 z^3 - 5x^2 y - 5$  e  $h(x, y) = x^3 - 0.7x^2 - 5xy$  são funções que a partir dos parâmetros de entrada retornam um valor calculado segundo as respectivas fórmulas, faça:

- 1) Em um script R implemente uma função chamada **calculaF**, que retorna o valor da  $f(x, y, z)$  e uma função chamada **calculaH** que retorna o valor da  $h(x, y)$ . A seguir, no mesmo script, implemente um algoritmo que apresenta na tela os valores retornados pelas funções **calculaF** e **calculaH** dados como parâmetros a função: **calculaF**, os valores **3**, **-2** e **-7** para **x**, **y** e **z**, respectivamente; **calculaH** os valores **-3** e **-4** para **x** e **y**, respectivamente. (2,0 pontos)

2) Elabore um script em R que cria os seguintes vetores:

- vetorX, contendo os elementos **1, -4, 5, -9, 8, 1, -3**, nesta ordem;
- vetorY, contendo uma sequência de valores entre **-4** e **2**, com intervalo de **1**;
- vetorZ, contendo **7** elementos iniciados em **-1.5**, com intervalo de **0.7**.

A seguir, no mesmo script, faça uso de um dos comandos do R que permitem criar ciclos para chamar a função **calculaF** utilizando como parâmetros trincas de dados (**x, y, z**) formadas pelos elementos dos vetores vetorX, vetorY e vetorZ. O script deve apresentar na tela o maior valor e a média, nesta ordem, dos valores retornados pela função **calculaF** para cada uma das trincas de dados. Observe que a primeira trinca de dados (**x, y, z**) possui os valores (**1, -4, -1.5**) e a segunda os valores (**-4, -3, -0.8**). (4,0 pontos)

4) Com base no fluxograma, escreva uma função chamada **processaFluxograma** que recebe como parâmetros os valores que deveriam, segundo o fluxograma, ser fornecidos pelo usuário, e, em vez de apresentar na tela o resultado do processamento, este seja retornado pela função. Teste a função utilizando os valores **4, 3** e **-1** para **x, y** e **z**, respectivamente. (4,0 pontos)

## Solução:

### Questão 1

```
calculaF <- function(x, y, z)

{

  retorno <- (x^3)*(z^3) - 5*(x^2)*y - 5

  return(retorno)

}
```

```
calculaH <- function(x, y)

{

  retorno <- (x^3) - 0.7*(x^2) - 5*x*y

  return(retorno)

}
```

#variáveis

```
valorX <- 0.0
```

```
valorY <- 0.0
```

```
valorZ <- 0.0
```

```
valor <- 0.0
```

#inicio

```
valorX <- 3
```

```
valorY <- -2
```

```
valorZ <- -7
```

```
valor <- calculaF(valorX, valorY, valorZ)
```

```
print(valor)
```

```
valorX <- -3
```

```
valorY <- -4
```

```
valor <- calculaH(valorX, valorY)
```

```
print(valor)
```

```
#fim
```

## Questão 2

```
calculaF <- function(x, y, z)
```

```
{
```

```
  retorno <- (x^3)*(z^3) - 5*(x^2)*y - 5
```

```
  return(retorno)
```

```
}
```

```
#variáveis
```

```
contador <- 0
```

```
vetorX <- NULL
```

```
vetorY <- NULL
```

```
vetorZ <- NULL
```

```
vetorValor <- NULL
```

```
#inicio
```

```
vetorX <- c(1, -4, 5, -9, 8, 1, -3)
```

```
vetorY <- -4:2
```

```
vetorZ <- seq(-1.5, by = 0.7, length = 7)
```

```
vetorValor <- rep(0, length(vetorX))
```

```
while (contador < length(vetorX))  
  
{  
  
  contador <- contador + 1  
  
  vetorValor[contador] <- calculaF(vetorX[contador], vetorY[contador], vetorZ[contador])  
  
}  
  
print(max(vetorValor))  
  
print(mean(vetorValor))  
  
#fim
```

### Questão 3

```
calculaF <- function(x, y, z)  
  
{  
  
  retorno <- (x^3)*(z^3) - 5*(x^2)*y - 5  
  
  return(retorno)  
  
}
```

```
calculaH <- function(x, y)  
  
{  
  
  retorno <- (x^3) - 0.7*(x^2) - 5*x*y  
  
  return(retorno)  
  
}
```

```
processaFluxograma <- function (x, y, z)  
  
{  
  
  retorno <- 0.0  
  
  peso <- 1
```

```
if (x >= 0) {  
  
  if (z < 0) {  
  
    retorno <- calculaH(x, y)  
  
  } else {  
  
    peso <- peso + 1  
  
    if (y > 0) {  
  
      peso <- peso + 2  
  
      retorno <- calculaF(x, y, z) + peso * calculaH(x, y)  
  
    } else {  
  
      retorno <- peso * calculaF(x, y, z) + calculaH(x, y)  
  
    }  
  
  }  
  
} else {  
  
  if (y > 0) {  
  
    peso <- peso + 2  
  
    retorno <- calculaF(x, y, z) + peso * calculaH(x, y)  
  
  } else {  
  
    retorno <- peso * calculaF(x, y, z) + calculaH(x, y)  
  
  }  
  
}  
  
return(retorno)  
  
}
```

#variáveis

x <- 0.0

```
y <- 0.0
```

```
z <- 0.0
```

```
valor <- 0.0
```

```
#inicio
```

```
x <- 4
```

```
y <- 3
```

```
z <- -1
```

```
valor <- processaFluxograma(x, y, z)
```

```
print(valor)
```

```
#fim
```