

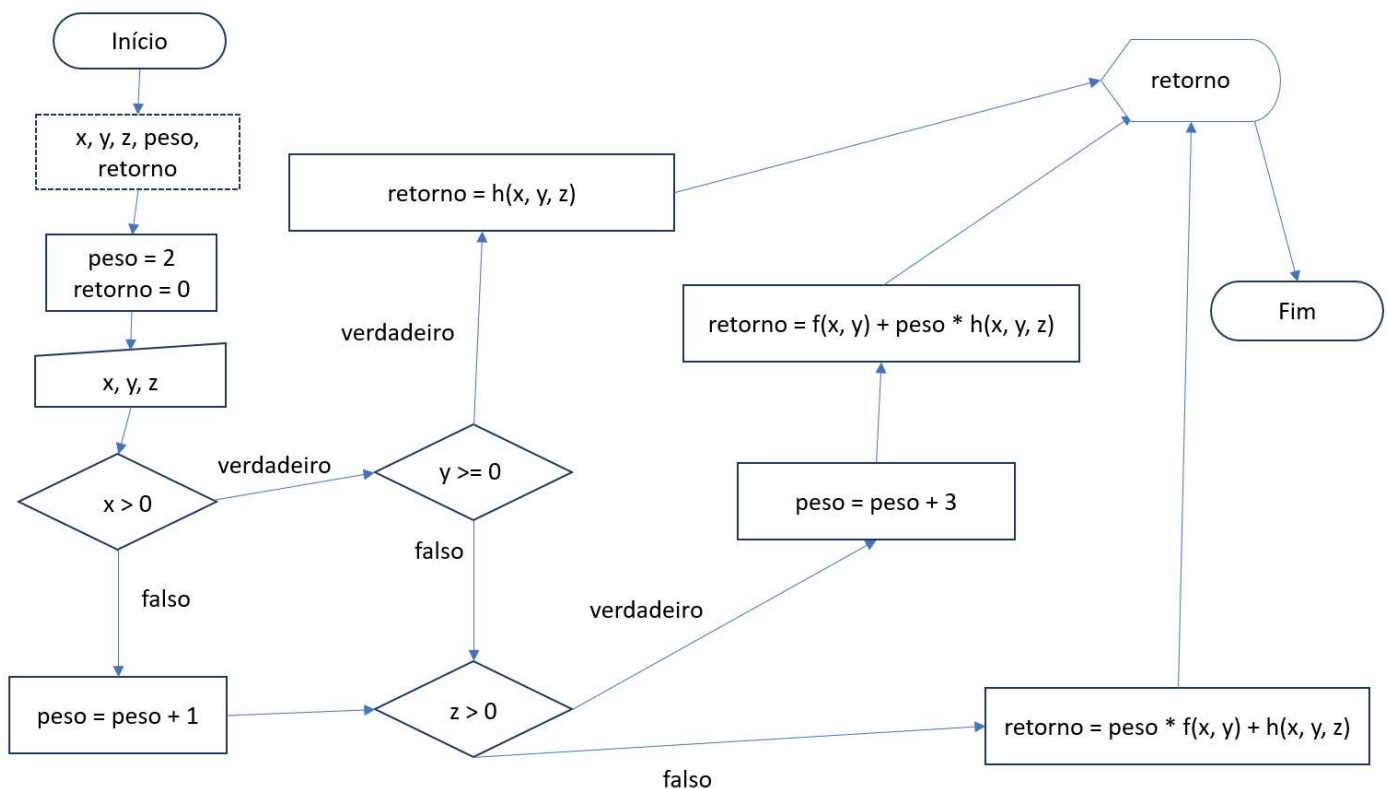
Nome: _____ Matrícula: _____ IP: _____

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 $f(x, y) = -3x^3 + x^2 + 5.7y$ e $h(x, y, z) = x^2y^4 - 7x^3z^2 + 5$ 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 $f(x, y)$ e uma função chamada **calculaH** que retorna o valor da $h(x, y, z)$. 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** e **-2** para x e y , respectivamente; **calculaH** os valores **2**, **-5** e **3** para x , y e z , respectivamente. (2,0 pontos)

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

- vetorX, contendo **7** elementos iniciados em **2**, com intervalo de **-0.8**;
- vetorY, contendo os elementos **3, -5, 7, -9, 5, 1, -3**, nesta ordem;
- vetorZ, contendo uma sequência de valores entre **-2** e **4**, com intervalo de **1**.

A seguir, no mesmo script, faça uso de um dos comandos do R que permitem criar ciclos para chamar a função **calculaH** utilizando como parâmetros trincas de dados (**x, y, z**) formadas pelos elementos dos vetores vetorX, vetorY e vetorZ. O script deve apresentar na tela a soma e a média, nesta ordem, dos valores retornados pela função **calculaH** para cada uma das trincas de dados. Observe que a primeira trinca de dados (**x, y, z**) possui os valores (**2, 3, -2**) e a segunda os valores (**1.2, -5, -1**). (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 **2, -3** e **1** para **x, y** e **z**, respectivamente. (4,0 pontos)

Solução:

Questão 1

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

```
{  
  
  retorno <- -3 * (x^3) + (x^2) + 5.7 * y  
  
  return(retorno)  
}
```

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

```
{  
  
  retorno <- (x^2) * (y^4) - 7 * (x^3) * (z^2) + 5  
  
  return(retorno)  
}
```

```
#variáveis
```

```
valorX <- 0.0
```

```
valorY <- 0.0
```

```
valorZ <- 0.0
```

```
valor <- 0.0
```

Questão 2

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

```
{  
  
  retorno <- (x^2) * (y^4) - 7 * (x^3) * (z^2) + 5  
  
  return(retorno)  
}
```

```
#variáveis
```

```
contador <- 0
```

```
vetorX <- NULL
```

```
vetorY <- NULL
```

```
vetorZ <- NULL
```

```
vetorValor <- NULL
```

```
#inicio
```

```
vetorX <- seq(2, by = -0.8, length = 7)
```

```
vetorY <- c(3, -5, 7, -9, 5, 1, -3)
```

```
vetorZ <- -2:4
```

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

```
while (contador < length(vetorX))
```

```
{
```

```
  contador <- contador + 1
```

```
  vetorValor[contador] <- calculaH(vetorX[contador], vetorY[contador], vetorZ[contador])
```

```
}
```

```
print(sum(vetorValor))
```

```
print(mean(vetorValor))
```

```
#fim
```

Questão 3

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

```
{
```

```
  retorno <- -3 * (x^3) + (x^2) + 5.7 * y
```

```
  return(retorno)
```

```
}
```

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

```
{
```

```
  retorno <- (x^2) * (y^4) - 7 * (x^3) * (z^2) + 5
```

```
  return(retorno)
```

```
}
```

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

```
{
```

```
  retorno <- 0.0
```

```
  peso <- 2
```

```
  if (x > 0) {
```

```
    if (y >= 0) {
```

```
      retorno <- calculaH(x, y, z)
```

```
    } else {
```

```
      if (z > 0) {
```

```
        peso <- peso + 3
```

```
        retorno <- calculaF(x, y) + peso * calculaH(x, y, z)
```

```
      } else {
```

```
        retorno <- peso * calculaF(x, y) + calculaH(x, y, z)
```

```
      }
```

```
    }
```

```
  } else {
```

```
    peso <- peso + 1
```

```
    if (z > 0) {
```

```
      peso <- peso + 3
```

```
    retorno <- calculaF(x, y) + peso * calculaH(x, y, z)

  } else {

    retorno <- peso * calculaF(x, y) + calculaG(x, y, z)

  }

}
```



```
return(retorno)

}
```

#variáveis

```
x <- 0.0
```

```
y <- 0.0
```

```
z <- 0.0
```

```
valor <- 0.0
```

#inicio

```
x <- 2
```

```
y <- -3
```

```
z <- 1
```

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

```
print(valor)
```

#fim