Introdução à Análise de Dados – Teste 2 (Modelo 1) – 2JA G1 – 2023.1

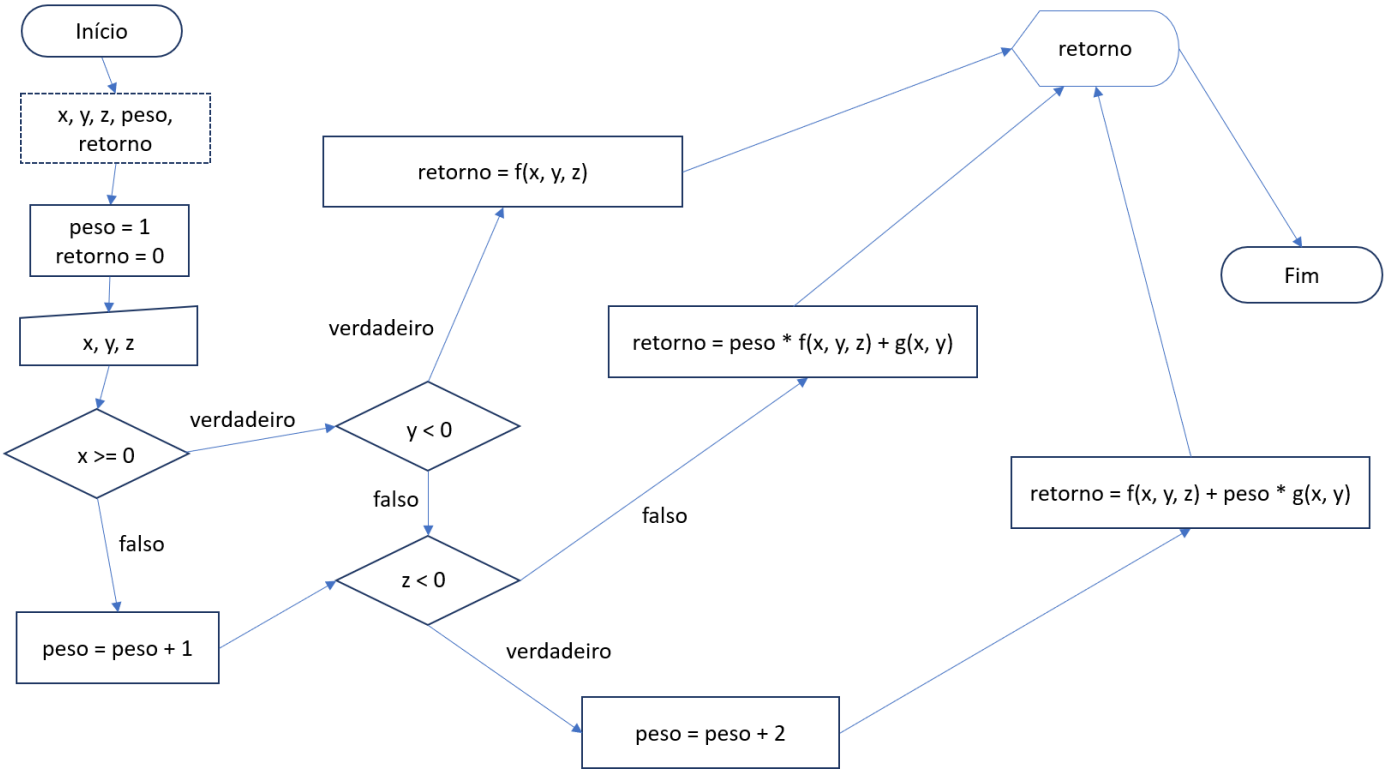
Nome:_____ **Matrícula:**_____ **IP:**_____

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 $f(x,y,z) = x^4y - 5x^2z^3 - 2$ e $g(x,y) = 0.9x^3 - 3x^2 + 5y$ 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 **calculaG** que retorna o valor da $g(x,y)$. A seguir, no mesmo script, implemente um algoritmo que apresenta na tela os valores retornados pelas funções **calculaF** e **calculaG** dados como parâmetros a função: **calculaF**, os valores **2**, **3** e **4** para $\boldsymbol{x}$, $\boldsymbol{y}$ e $\boldsymbol{z}$, respectivamente; **calculaG** os valores **7** e **-2** para $\boldsymbol{x}$ e $\boldsymbol{y}$, respectivamente. (2,0 pontos)

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

- vetorX, contendo uma sequência de valores entre -**3** e **3**, com intervalo de **1**;
- vetorY, contendo os elementos **2**, **4**, **5**, -**6**, **8**, -**3**, **1**, nesta ordem.
- vetorZ, contendo **7** elementos iniciados em **-1**, 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 ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$) 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 **calculaF** para cada uma das trincas de dados. Observe que a primeira trinca de dados ($\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$) possui os valores (-**3**, **2**, **-1**) e a segunda os valores (**-2**, **4**, **-0.3**). (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 **-3**, **2** e **-1** para $\boldsymbol{x}$, $\boldsymbol{y}$ e $\boldsymbol{z}$, respectivamente. (4,0 pontos)

Soluções:

Questão 1

```
calculaF <- function(x, y, z)
{
  retorno <- (x^4)*y - 5*(x^2)*(z^3) - 2
  return(retorno)
}
calculaG <- function(x, y)
{
  retorno <- 0.9*(x^3) - 3*(x^2) + 5*y
  return(retorno)
}

#variáveis
valorX <- 0.0
valorY <- 0.0
valorZ <- 0.0
valor <- 0.0

#inicio
valorX <- 2
valorY <- 3
valorZ <- 4

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

valorX <- 7
valorY <- -2

valor <- calculaG(valorX, valorY)
print(valor)

#fim
```

Questão 2

```
calculaF <- function(x, y, z)
{
  retorno <- (x^4)*y - 5*(x^2)*(z^3) - 2
  return(retorno)
}

#variáveis
contador <- 0
vetorX <- NULL
vetorY <- NULL
vetorZ <- NULL
vetorValor <- NULL

#inicio
vetorX <- -3:3
vetorY <- c(2, 4, 5, -6, 8, -3, 1)
vetorZ <- seq(-1, 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(sum(vetorValor))
print(mean(vetorValor))
#fim

calculaF <- function(x, y, z)
{
  retorno <- (x^4)*y - 5*(x^2)*(z^3) - 2
  return(retorno)
```

```
}

calculaG <- function(x, y)
{
  retorno <- 0.9*(x^3) - 3*(x^2) + 5*y
  return(retorno)
}

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

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

```
  return(retorno)
}

#variáveis
x <- 0.0
y <- 0.0
z <- 0.0
valor <- 0.0

#inicio
x <- -3
y <- 2
z <- -1

valor <- processaFluxograma(x, y, z)
print(valor)
#fim
```

Questão 3

```
calculaF <- function(x, y, z)
{
  retorno <- (x^4)*y - 5*(x^2)*(z^3) - 2
  return(retorno)
}

calculaG <- function(x, y)
{
  retorno <- 0.9*(x^3) - 3*(x^2) + 5*y
  return(retorno)
}

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

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

```
    retorno <- calculaF(x, y, z) + peso * calculaG(x, y)
   } else {
    retorno <- peso * calculaF(x, y, z) + calculaG(x, y)
   }
  }

  return(retorno)
}

#variáveis
x <- 0.0
y <- 0.0
z <- 0.0
valor <- 0.0

#inicio
x <- -3
y <- 2
z <- -1

valor <- processaFluxograma(x, y, z)
print(valor)
#fim
```