

TESTE 3 (INDIVIDUAL OU EM DUPLA)

INF-0612 – ANÁLISE DE DADOS

O objetivo deste teste é exercitar os conceitos vistos até agora na disciplina. São quatro problemas a serem resolvidos:

(2.5 pontos) Máximo Divisor Comum

Implemente uma função com a seguinte assinatura `gcd(...)` que, dados dois ou mais números inteiros não negativos, calcule o Máximo Divisor Comum (Greatest Common Divisor) entre eles.

Observações:

- Você deve utilizar a função `gcd2(x, y)`, que foi vista em sala de aula, como sub-rotina da função `gcd(...)`.
- A implementação da função `gcd2(x, y)` está disponível no arquivo `teste3.R`.

Exemplos de saídas esperadas para a função:

```
1 > gcd(15, 21, 36)
2 [1] 3
3
4 > gcd(4, 8, 16, 17)
5 [1] 1
```

(2.5 pontos) Moda da Idade da Turma

Você foi encarregado de criar uma programa que, ao receber um vetor de inteiros representando a idade dos alunos de uma turma, calcule a moda desse vetor.

Definição de moda (Mo): valor(es) que ocorre(m) com maior frequência em um conjunto de dados.

Por exemplo, considere o vetor `idade = {21, 18, 18, 19, 18}` que representa a idade de cinco alunos de uma turma. A moda desse conjunto de dados será a idade que mais aparece, ou seja: `Mo = 18`. Implemente uma função com a seguinte assinatura `mode(vector)` que, dado um vetor de inteiros, calcule a moda do vetor dado como entrada.

Observações:

- Você deve utilizar a função `count(vector, element)`, que foi vista em sala de aula, como sub-rotina da função `mode(vector)`.
- A implementação da função `count(vector, element)` está disponível no arquivo `teste3.R`.

Exemplos de saídas esperadas para a função:

```
1 > mode(sample(5))
2 [1] 3 5 4 1 2
3
4 > mode(c(18, 19, 21, 19, 18, 19, 18))
5 [1] 18 19
6
7 > mode(c(18, 19, 17, 19, 17, 17))
8 [1] 17
```

(2.5 pontos) Binário para Decimal

Implemente uma função com a seguinte assinatura `binToDec(...)`. Essa função receberá como parâmetros uma quantidade indefinida de vetores binários. Cada vetor de binários simboliza a representação binária de um número inteiro. Por exemplo, o vetor `c(1, 0, 1)` representa o número $5 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$. A saída esperada para essa função deve ser um vetor de inteiros onde a primeira posição desse vetor é representação decimal do primeiro vetor de binários dado como entrada, a segunda posição do vetor é a representação decimal do segundo vetor de binários dado como entrada, e assim por diante.

Exemplos de saídas esperadas para a função:

```
1 > binToDec(c(1, 0))
2 [1] 2
3
4 > binToDec(c(0, 0, 1), c(1, 1))
5 [1] 1 3
6
7 > binToDec(rep(1, 3), rep(0, 2), rep(c(1,0), 2))
8 [1] 7 0 10
```

(2.5 pontos) Ocorrência de Palavras

A repetição de uma mesma palavra em um texto não é uma boa prática de escrita. Imagine que você é um(a) funcionário(a) de uma grande empresa de publicidade. Atualmente, na sua empresa, o processo que contabiliza a ocorrência de um determinada palavra em um texto é feito de forma manual por funcionários. Seu gerente quer automatizar esse processo, e deixou você encarregado de criar um mecanismo para tal. Implemente uma função com a seguinte assinatura `wordCount(word, text)`, que recebe dois parâmetros, ambos são tipo **character** (String), o primeiro deles (**word**) representa a palavra e o segundo (**text**) representa o texto. A saída dessa função deve ser um número inteiro que determina a quantidade de ocorrências da palavra no texto que foram dados como parâmetros.

Observações:

- Você deve implementar a função de maneira *case insensitive*, ou seja, uma mesma palavra pode estar escrita em caixa alta ou caixa baixa, mas devem ser consideradas iguais.
- Você pode assumir que o texto e a palavra não terão acentos.
- Você pode assumir que o texto só apresentará as seguintes pontuações: “.”, “,”, “!” e “?”.

Exemplos de saídas esperadas para a função:

```
1 > text <- "O rAto roeu a roupa do Rei de Roma! RainhA raivosa rasgou o resto."
2 > wordCount("rato", text)
3 [1] 1
4 > wordCount("roma", text)
5 [1] 1
6
7 > text <- "A vaca malHada foi molhADA por outra VACA, MOLhada e MALhaDa."
8 > wordCount("outra", text)
9 [1] 1
10 > wordCount("vaca", text)
11 [1] 2
12 > wordCount("malhada", text)
13 [1] 2
14
15 > text <- "Se a liga me ligasse, eu tambem ligava a liga. Mas a liga nao me
16   liga, eu tambem nao ligo a liga."
17 > wordCount("liga", text)
18 [1] 5
19 > wordCount("ligasse", text)
[1] 1
```

Considerações Finais

- A utilização de qualquer função ou estrutura não vista em sala de aula, ou seja, que não encontram-se nos slides fornecidos no Moodle, acarretará em um desconto de 50% da nota daquele item.
- Você não deve remover qualquer linha já existente no arquivo.
- Para testes realizados em dupla: apenas um membro da dupla deve enviar a solução. Os nomes dos membros devem constar no cabeçalho do arquivo .R.
- Salve o arquivo utilizando o mesmo nome, e o envie no sistema Moodle, clicando no link “**Teste 3**” da Seção “**Avaliações**”. Clique em “**Adicionar tarefa**”, anexe o arquivo e, por fim, clique em “**Salvar mudanças**”. Você voltará para a tela da atividade e deverá constar o status “**Enviado para avaliação**”. A qualquer momento, antes do prazo final de submissão, você pode alterar sua submissão clicando em “**Editar envio**”.

Prazo de entrega: 22 de março de 2018 (quinta-feira) às 23:55hs.

Forma de entrega: via sistema Moodle:

- <https://moodle.lab.ic.unicamp.br/moodle/course/view.php?id=252>

Pontuação: Este teste será pontuado de 0 a 10, e corresponderá 20% da nota final.