

Atividade 10

Valor: 3,0 pontos na Segunda Nota

Primeira Chamada: 02/12/2025

Segunda Chamada: 03/12/2025.

Questão 01 (1,0 ponto)

Considere um arquivo `notas.txt` onde cada linha representa um aluno e suas três notas, separadas por dois pontos (:):

```
# Conteúdo de notas.txt
Aluno:Nota1:Nota2:Nota3
João Marcelo:10.0:10.0:10.0
Alisson Barbosa:1.0:2.0:3.0
Jeandro Bezerra:7.0:8.0:9.0
Marcos Dantas:5.0:4.0:6.0
Michel Sales:3.0:4.0:2.0
```

Crie um *script* AWK chamado `disciplina.awk` para gerar um relatório completo.

Requisitos do Script:

1. **Separador de Campo (FS)**: O script deve ser invocado usando o separador de campo `-F:` (dois pontos).
2. **Cabeçalho (BEGIN)**: Use o padrão `BEGIN` para imprimir o cabeçalho: `Aluno:Situação:Média`.
3. **Cálculo da Média**: Para cada aluno, calcule a média simples das três notas (`$2` , `$3` , `$4`).
4. **Situação do Aluno (if/else)**: Use estruturas condicionais (`if/else`) dentro do bloco de ação para determinar a situação:
 - Média ≥ 7.0 : **Aprovado**
 - $5.0 \leq \text{Média} < 7.0$: **Final**
 - Média < 5.0 : **Reprovado**
5. **Média Geral das Provas (END)**: Use o padrão `END` para, ao final do arquivo, calcular e imprimir a média de cada prova (Nota1, Nota2, Nota3) considerando **todos** os alunos. Use a variável interna `NR` (Número de Registros) no cálculo.

Resultado Esperado:

Bash

```
$ awk -F: -f disciplina.awk notas.txt
Aluno:Situação:Média
João Marcelo:Aprovado:10.0
Alisson Barbosa:Reprovado:2.0
Jeandro Bezerra:Aprovado:8.0
Marcos Dantas:Final:5.0
```

Michel Sales:Reprovado:3.0
Média das Provas: 5.2 5.6 6.0

Questão 02 (1,0 ponto)

O objetivo é processar dados de IP e latência e ordenar o resultado final. Diferente do Shell Script, o AWK deve usar sua capacidade de redirecionar a saída para um comando externo (`pipe`) para realizar a ordenação.

Considere que o arquivo `ips_latencia.txt` já contém as latências médias calculadas (você pode criar este arquivo com o conteúdo abaixo para testar):

```
# Conteúdo de ips_latencia.txt (separador é ESPAÇO)
192.168.0.1 11.1
54.230.57.207 55.4
8.8.8.8 94.0
```

Crie um *script* AWK chamado `latencia.awk` que execute o seguinte:

1. **Armazenamento em Vetor:** Para cada linha lida, armazene o endereço IP (`$1`) e o valor da Latência (`$2`) em um **vetor (array)**, usando o IP como chave.
2. **Processamento Final (END):** No bloco `END` , itere sobre o vetor usando o comando `for` (`chave in vetor`) .
3. **Ordenação com Pipe:** Redirecione a saída de cada item (`print`) para o comando externo `sort` através de um *pipe*, para ordenar o resultado pelo valor da latência (o segundo campo), do menor para o maior. *Dica:* Use `| "sort -k2n"` .
4. **Formatação de Saída:** Use `printf` ou manipulação de *string* para garantir que a unidade `ms` seja adicionada ao valor da latência na saída final.

Resultado Esperado:

Bash

```
$ awk -f latencia.awk ips_latencia.txt
192.168.0.1 11.1ms
54.230.57.207 55.4ms
8.8.8.8 94.0ms
```

Questão 03 - 1,0 ponto

Você recebeu um arquivo de dados de vendas chamado `vendas.txt` , onde os campos são separados por ponto e vírgula (`;`). O formato de cada linha é: `Produto;PrecoUnitario;Quantidade` .

```
# Conteúdo de exemplo vendas.txt
Teclado;50;10
Mouse;20;5
Monitor;800;2
Cabo HDMI;15;3
Webcam;150;4
```

Desenvolva um *script* AWK chamado `faturamento.awk` que realize as seguintes tarefas:

1. **Cálculo por Linha:** Para cada produto, calcule o **Valor Total** (Preço Unitário multiplicado pela Quantidade).
2. **Filtragem:** Imprima o nome do produto e o seu valor total, mas **apenas** para os produtos cujo valor total seja **maior ou igual a 200**.
3. **Formatação:** A saída deve ser exibida no formato `Produto: ValorTotal`.

Exemplo de execução e resultado esperado:

Bash

```
$ awk -F\; -f faturamento.awk vendas.txt
Teclado: 500
Monitor: 1600
Webcam: 600
```

(Observe que itens como "Mouse" e "Cabo HDMI" não aparecem pois seus totais são inferiores a 200).