



Introdução à Ciência da Computação – Lista 6

Shell script – parte 3

Nome: RA:

- 1) Crie um script chamado scriptaritmetico, com uma operação aritmética arbitrária usando pelo menos 4 variáveis, realizando uma operação de divisão cujo resultado não seja um número inteiro. Execute o script e mostre o resultado. Qual o recurso a ser utilizado caso você queira que o valor não inteiro apareça no resultado? Qual variável eu uso para isso?

```
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod +x scriptaritmetico.sh
./scriptaritmetico.sh
Resultado: 1.12
2025.1.08.027@suporte-OptiPlex-3050:~$
```

```
1 #!/bin/bash
2 a=7
3 b=2
4 c=3
5 d=5
6
7 resultado=$(echo "scale=2; ($a + $b) / ($c + $d)" | bc)
8 echo "Resultado: $resultado"
9
```

Comando | bc

Scale=2

- 2) Ponha em execução a calculadora bc. Mostre o uso da variável scale, exibindo um resultado de operação aritmética com 6 casas decimais.

```
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod +x scriptaritmetico.sh
./scriptaritmetico.sh
Resultado: 1.12
2025.1.08.027@suporte-OptiPlex-3050:~$ bc -q
scale=6
7/2
3.500000
```

- 3) Crie um script simples chamado testebc, em que você utilize a calculadora bc dentro dele, envolvendo o uso de algumas variáveis e a operação de divisão, com o direcionamento via pipe. Execute o script, mostrando o resultado.

```
2025.1.08.027@suporte-OptiPlex-3050:~$ gedit testebc.sh
^C
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod +x testebc.sh
./testebc.sh
3.3333
2025.1.08.027@suporte-OptiPlex-3050:~$
```

- 4) Crie um script chamado testebccomplexo, em que você utilize operações aritméticas diversas com a calculadora bc (pelo menos duas), armazenando os resultados em variáveis, como mostrado na aula. Neste caso, utilize a técnica de redirecionamento de entrada inline. Execute o script, mostrando o resultado.

```
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo.sh
./testebccomplexo.sh
Resultado 1: 37
Resultado 2: 3.333
2025.1.08.027@suporte-OptiPlex-3050:~$
```

- 5) O que consiste o status de saída de um programa?

O status de saída de um programa, também conhecido como código de retorno ou código de saída, é um valor numérico que indica o resultado da execução desse programa. Zero quando é sucesso e um número maior que zero para erro.

Mostre um exemplo de execução de dois comandos (um com sucesso e outro desconhecido) e verifique esse status. Mostre em tela.

```
2025.1.08.027@suporte-OptiPlex-3050:~$ ls
echo "Status de ls: $?"

comando_inexistente
echo "Status de comando_inexistente: $?"
Desktop      intcc        Public       Templates
Documents    Music        scriptaritmetico.sh  testebccomplexo.sh
Downloads    Pictures     snap         Videos
Status de ls: 0
comando_inexistente: command not found
Status de comando_inexistente: 127
```

- 6) Qual a função do comando exit?

Por padrão, seu shell script finaliza com o status de saída do último comando executado no script. Utilizado para finalizar a execução de um script ou de um shell, retornando um código de saída específico. Mostre um exemplo do uso do comando exit dentro de um script, mudando o valor padrão do status de saída.

Mostre tanto o uso do exit exibindo um número qualquer até 255, quanto o valor de uma variável que você utilize no script. Execute o script e mostre o valor do status de saída em cada caso.

```

2025.1.08.027@suporte-OptiPlex-3050:~$ gedit ativ6.sh
^C
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod a+x ativ6.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ ./ativ6.sh
Saíndo com status 5...
2025.1.08.027@suporte-OptiPlex-3050:~$ $?
5: command not found
2025.1.08.027@suporte-OptiPlex-3050:~$ gedit ativ6.sh
^C
2025.1.08.027@suporte-OptiPlex-3050:~$ ./ativ6.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ $?
33: command not found
2025.1.08.027@suporte-OptiPlex-3050:~$ █

```

- 7) Crie um script simples envolvendo comandos condicionais if then else, para verificar a existência de um diretório específico no seu home. Primeiro procure um diretório inexistente, depois um diretório existente e exiba as mensagens específicas de acordo com o resultado. Execute o script e mostre em tela.

```

2025.1.08.027@suporte-OptiPlex-3050:~$ gedit ativ7.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod a+x ativ7.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ ./ativ7.sh
O diretório /home/2025.1.08.027/diretorio_teste não existe.
2025.1.08.027@suporte-OptiPlex-3050:~$

```

- 8) Crie um script envolvendo várias condicionais usando a estrutura if then elif else, fazendo duas operações aritméticas arbitrárias, verificando o valor das variáveis que armazenam essa operação, checando se o valor da primeira é maior, menor ou igual ao valor da segunda. Execute o script e mostre o resultado em tela.

```

2025.1.08.027@suporte-OptiPlex-3050:~$ gedit ativ8.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod a+x ativ8.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ ./ativ8.sh
A > B
2025.1.08.027@suporte-OptiPlex-3050:~$

```

- 9) Crie um script envolvendo condicionais usando a estrutura if then else, criando duas variáveis string arbitrárias e verificando seus valores, checando se o conteúdo das variáveis é igual. Execute o script e mostre o resultado em tela.

```

2025.1.08.027@suporte-OptiPlex-3050:~$ gedit ativ9.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod a+x ativ9.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ ./ativ9.sh
As strings são iguais
2025.1.08.027@suporte-OptiPlex-3050:~$

```

- 10) Crie um script envolvendo condicionais usando a estrutura if then else, criando uma string com um conteúdo, verificando se seu valor é "fruta". Execute o script e mostre o resultado em

```

2025.1.08.027@suporte-OptiPlex-3050:~$ gedit ativ10.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod a+x ativ10.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ ./ativ10.sh
É uma fruta
2025.1.08.027@suporte-OptiPlex-3050:~$ █

```

tela.

- 11) Crie um script envolvendo condicionais usando a estrutura if then else, criando duas strings, uma vazia, outra com conteúdo e verificando estes resultados (se tem conteúdo em ambos

os casos).

```
2025.1.08.027@suporte-OptiPlex-3050:~$ gedit ativ11.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod a+x ativ11.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ ./ativ11.sh
str1 está vazia
str2 tem conteúdo
2025.1.08.027@suporte-OptiPlex-3050:~$
```

12) Cite 5 opções de comparações envolvendo arquivos. Escolha uma das opções e crie um script envolvendo essa opção.

-e arquivo → existe

-f arquivo → é um arquivo normal

-d arquivo → é um diretório

-r arquivo → tem permissão de leitura

-x arquivo → é executável

```
2025.1.08.027@suporte-OptiPlex-3050:~$ gedit ativ12.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ chmod a+x ativ12.sh
2025.1.08.027@suporte-OptiPlex-3050:~$ ./ativ12.sh
Arquivo não encontrado.
2025.1.08.027@suporte-OptiPlex-3050:~$
```