



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?

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ nano scriptaritmetico  
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x scriptaritmetico  
2025.1.08.033@suporte-OptiPlex-3050:~$ ./scriptaritmetico  
Resultado da operação é: 1.80  
2025.1.08.033@suporte-OptiPlex-3050:~$
```

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

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ bc  
bc 1.07.1  
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software  
Foundation, Inc.  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type 'warranty'.  
scale=6  
7 / 3  
2.333333  
quit  
2025.1.08.033@suporte-OptiPlex-3050:~$ echo "scale=6; 7/3" | bc  
2.333333  
2025.1.08.033@suporte-OptiPlex-3050:~$
```

- 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.

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ nano testebc  
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x testebc  
2025.1.08.033@suporte-OptiPlex-3050:~$ ./testebc  
Resultado da divisão de 10 por 3 é: 3.3333  
2025.1.08.033@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.

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ nano testebccomplexo  
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x testebccomplexo  
2025.1.08.033@suporte-OptiPlex-3050:~$ ./testebccomplexo  
Resultados das operações:  
17  
3.1428  
2025.1.08.033@suporte-OptiPlex-3050:~$
```

- 5) O que consiste o status de saída de um programa? Mostre um exemplo de execução de dois comandos (um com sucesso e outro desconhecido) e verifique esse status. Mostre em tela.

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ ls  
echo $?  
arquivo_log.txt      Music      Templates  
Desktop              NetBeansProjects testebc  
Documents             Pictures   testebccomplexo  
Downloads             Public     testecrases.sh  
exemplo_here.sh      scriptaritmetico testevariaveisambiente.sh  
listagem_20250523084525.txt scriptaritmetico.sh testevariaveis.sh  
meuscript.sh          snap      Videos  
0  
2025.1.08.033@suporte-OptiPlex-3050:~$ comandoinvalido  
echo $?  
comandoinvalido: command not found  
127  
2025.1.08.033@suporte-OptiPlex-3050:~$
```

- 6) Qual a função do comando exit? 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.

```
2025.1.08.033@suporte-OptiPlex-3050: ~
2025.1.08.033@suporte-OptiPlex-3050:~$ nano testexit
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x testexit
2025.1.08.033@suporte-OptiPlex-3050:~$ ./testexit
echo $?
Executando parte 1: exit com número fixo
42
2025.1.08.033@suporte-OptiPlex-3050:~$ nano testexit
2025.1.08.033@suporte-OptiPlex-3050:~$ ./testexit
echo $?
Executando parte 1: exit com número fixo
42
2025.1.08.033@suporte-OptiPlex-3050:~$ Executando parte 1: exit com número fixo
Executando parte 2: exit com variável
77
Executando: command not found
Executando: command not found
77: command not found
2025.1.08.033@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.

```
2025.1.08.033@suporte-OptiPlex-3050: ~
2025.1.08.033@suporte-OptiPlex-3050:~$ nano verificadiretorio
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x verificadiretorio
2025.1.08.033@suporte-OptiPlex-3050:~$ ./verificadiretorio
0 diretório /home/2025.1.08.033/diretorio_teste NÃO existe.
2025.1.08.033@suporte-OptiPlex-3050:~$ nano verificadiretorio
2025.1.08.033@suporte-OptiPlex-3050:~$ ./verificadiretorio
0 diretório /home/2025.1.08.033/Downloads EXISTE.
2025.1.08.033@suporte-OptiPlex-3050:~$ 0 diretório /home/seuusuario/Downloads EXISTE.
0: command not found
2025.1.08.033@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.

```
2025.1.08.033@suporte-OptiPlex-3050: ~
2025.1.08.033@suporte-OptiPlex-3050:~$ nano comparaaritmetica
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x comparaaritmetica
2025.1.08.033@suporte-OptiPlex-3050:~$ ./comparaaritmetica
Valor 1 = 8
Valor 2 = 8
Valor 1 é IGUAL a Valor 2
2025.1.08.033@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.

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ nano verifica_strings.sh  
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x verifica_strings.sh  
2025.1.08.033@suporte-OptiPlex-3050:~$ ./verifica_strings.sh  
As strings são diferentes.  
2025.1.08.033@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

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ nano verifica_fruta.sh  
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x verifica_fruta.sh  
2025.1.08.033@suporte-OptiPlex-3050:~$ ./verifica_fruta.sh  
A string é fruta.  
2025.1.08.033@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

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ nano verifica_strings.sh  
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x verifica_strings.sh  
2025.1.08.033@suporte-OptiPlex-3050:~$ ./verifica_strings.sh  
A string_vazia está vazia.  
A string_com_conteudo tem conteúdo.  
2025.1.08.033@suporte-OptiPlex-3050:~$
```

os casos).

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

```
2025.1.08.033@suporte-OptiPlex-3050: ~  
2025.1.08.033@suporte-OptiPlex-3050:~$ nano verifica_arquivo.sh  
2025.1.08.033@suporte-OptiPlex-3050:~$ chmod +x verifica_arquivo.sh  
2025.1.08.033@suporte-OptiPlex-3050:~$ ./verifica_arquivo.sh  
0 arquivo 'teste.txt' NÃO existe.  
2025.1.08.033@suporte-OptiPlex-3050:~$ echo "Conteúdo de exemplo" > teste.txt  
2025.1.08.033@suporte-OptiPlex-3050:~$ ./verifica_arquivo.sh  
0 arquivo 'teste.txt' existe.  
2025.1.08.033@suporte-OptiPlex-3050:~$
```