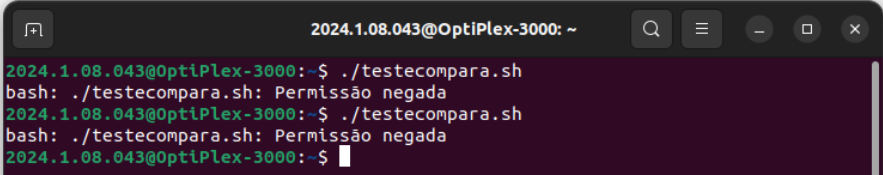


Bartholomeu vitor neto
2024.1.08.043

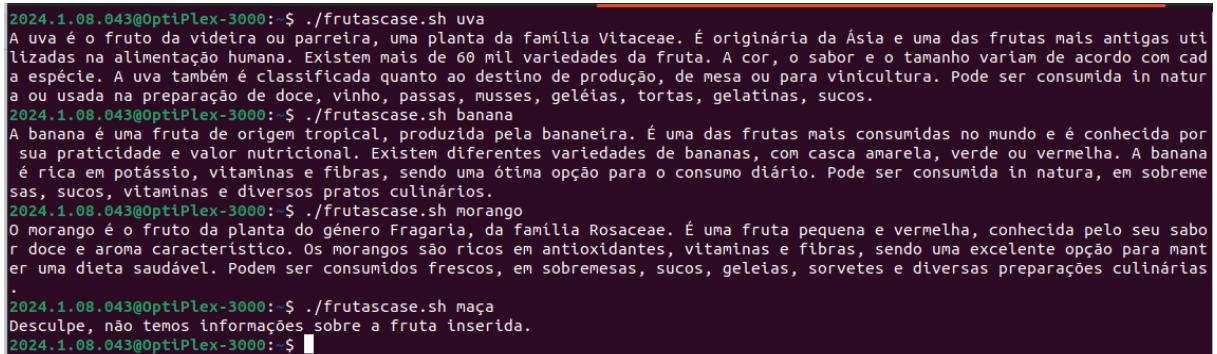
1.Crie um script chamado escrevenome, faça com que a saída desse script seja seu nome completo. Não utilize o comando chmod. Depois crie um script chamado testecompara, utilize o operador AND e verifique se o usuário logado tem permissão r e x sobre o script escrevenome. Mostre o resultado da saída.

```
1 [ -r escrevenome.sh ] && [ -x escrevenome.sh ] && echo "O usuário possui permissão" || echo "Permissão negada"
2
```



```
2024.1.08.043@OptiPlex-3000: ~
2024.1.08.043@OptiPlex-3000:~$ ./testecompara.sh
bash: ./testecompara.sh: Permissão negada
2024.1.08.043@OptiPlex-3000:~$ ./testecompara.sh
bash: ./testecompara.sh: Permissão negada
2024.1.08.043@OptiPlex-3000:~$
```

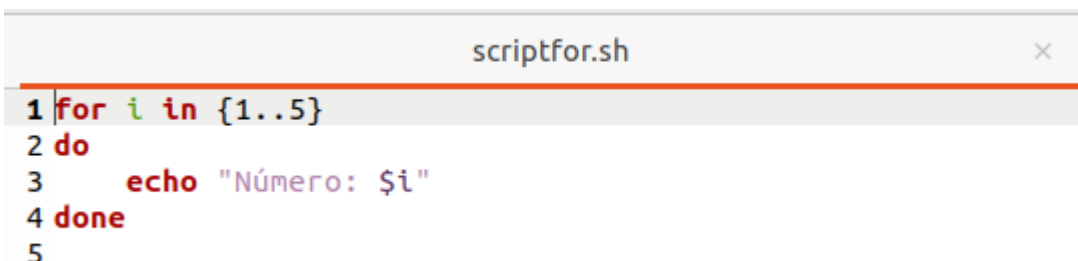
2.Crie um script chamado frutascase. Com base no valor da variável fruta mostre uma breve descrição da fruta. Faça com 5 frutas. Exemplo: fruta=uva, echo "A uva é o fruto da videira ou parreira, uma planta da família Vitaceae. É originária da Ásia e uma das frutas mais antigas utilizadas na alimentação humana. Existem mais de 60 mil variedades da fruta. A cor, o sabor e o tamanho variam de acordo com cada espécie. A uva também é classificada quanto ao destino de produção, de mesa ou para vinicultura. Pode ser consumida in natura ou usada na preparação de doce, vinho, passas, musses, geléias, tortas, gelatinas, sucos."



```
2024.1.08.043@OptiPlex-3000:~$ ./frutascase.sh uva
A uva é o fruto da videira ou parreira, uma planta da família Vitaceae. É originária da Ásia e uma das frutas mais antigas utilizadas na alimentação humana. Existem mais de 60 mil variedades da fruta. A cor, o sabor e o tamanho variam de acordo com cada espécie. A uva também é classificada quanto ao destino de produção, de mesa ou para vinicultura. Pode ser consumida in natura ou usada na preparação de doce, vinho, passas, musses, geléias, tortas, gelatinas, sucos.
2024.1.08.043@OptiPlex-3000:~$ ./frutascase.sh banana
A banana é uma fruta de origem tropical, produzida pela bananeira. É uma das frutas mais consumidas no mundo e é conhecida por sua praticidade e valor nutricional. Existem diferentes variedades de bananas, com casca amarela, verde ou vermelha. A banana é rica em potássio, vitaminas e fibras, sendo uma ótima opção para o consumo diário. Pode ser consumida in natura, em sobremesas, sucos, vitaminas e diversos pratos culinários.
2024.1.08.043@OptiPlex-3000:~$ ./frutascase.sh morango
O morango é o fruto da planta do gênero Fragaria, da família Rosaceae. É uma fruta pequena e vermelha, conhecida pelo seu sabor doce e aroma característico. Os morangos são ricos em antioxidantes, vitaminas e fibras, sendo uma excelente opção para manter uma dieta saudável. Podem ser consumidos frescos, em sobremesas, sucos, geleias, sorvetes e diversas preparações culinárias.
2024.1.08.043@OptiPlex-3000:~$ ./frutascase.sh maça
Desculpe, não temos informações sobre a fruta inserida.
2024.1.08.043@OptiPlex-3000:~$
```

3.Cite, explique e faça um script simples para cada estrutura de repetição do shell bash. Use sua criatividade para os scripts.

FOR : é usado para iterar sobre uma lista de elementos e executar um bloco de código para cada elemento na lista.



```
scriptfor.sh
1 for i in {1..5}
2 do
3     echo "Número: $i"
4 done
5
```

WHILE: executa um bloco de código repetidamente enquanto uma condição específica for verdadeira.

```
scriptfor.sh  x  scriptwhile.sh  x
1 count=1
2
3 while [ $count -le 5 ]
4 do
5     echo "Contagem: $count"
6     ((count++))
7 done
```

UNTIL é semelhante ao WHILE, mas ele executa um bloco de código repetidamente enquanto uma condição específica for falsa.

logo abaixo temos também a saída , junto com o script until.

```
2024.1.08.043@OptiPlex-3000:~$ chmod +x scriptfor.sh
2024.1.08.043@OptiPlex-3000:~$ ./scriptfor.sh
Número: 1
Número: 2
Número: 3
Número: 4
Número: 5
2024.1.08.043@OptiPlex-3000:~$ chmod +x scriptwhile.sh
2024.1.08.043@OptiPlex-3000:~$ ./scriptwhile.sh
Contagem: 1
Contagem: 2
Contagem: 3
Contagem: 4
Contagem: 5
2024.1.08.043@OptiPlex-3000:~$ chmod +x scriptuntil.sh
2024.1.08.043@OptiPlex-3000:~$ ./scriptuntil.sh
Contagem: 1
Contagem: 2
Contagem: 3
Contagem: 4
Contagem: 5
2024.1.08.043@OptiPlex-3000:~$ █
```

Abrir ▾

+

scriptuntil.sh

~/

Salvar

≡

—

□

×

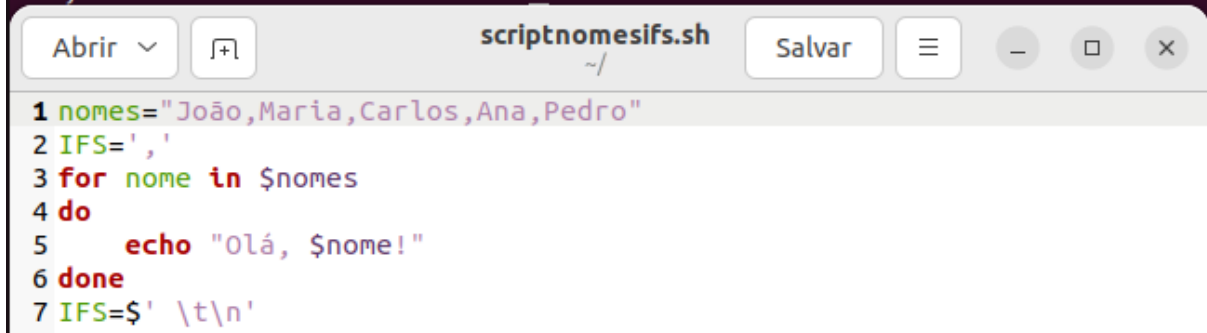
scriptfor.sh x scriptwhile.sh x scriptuntil.sh x

```
1 count=1
2
3 until [ $count -gt 5 ]
4 do
5     echo "Contagem: $count"
6     ((count++))
7 done
```

sh ▾ Largura da tabulação: 8 ▾ Lin 1, Col 1 ▾ INS

4.Explique o que é IFS e faça um script diferente do que foi visto em aula. Use sua criatividade.

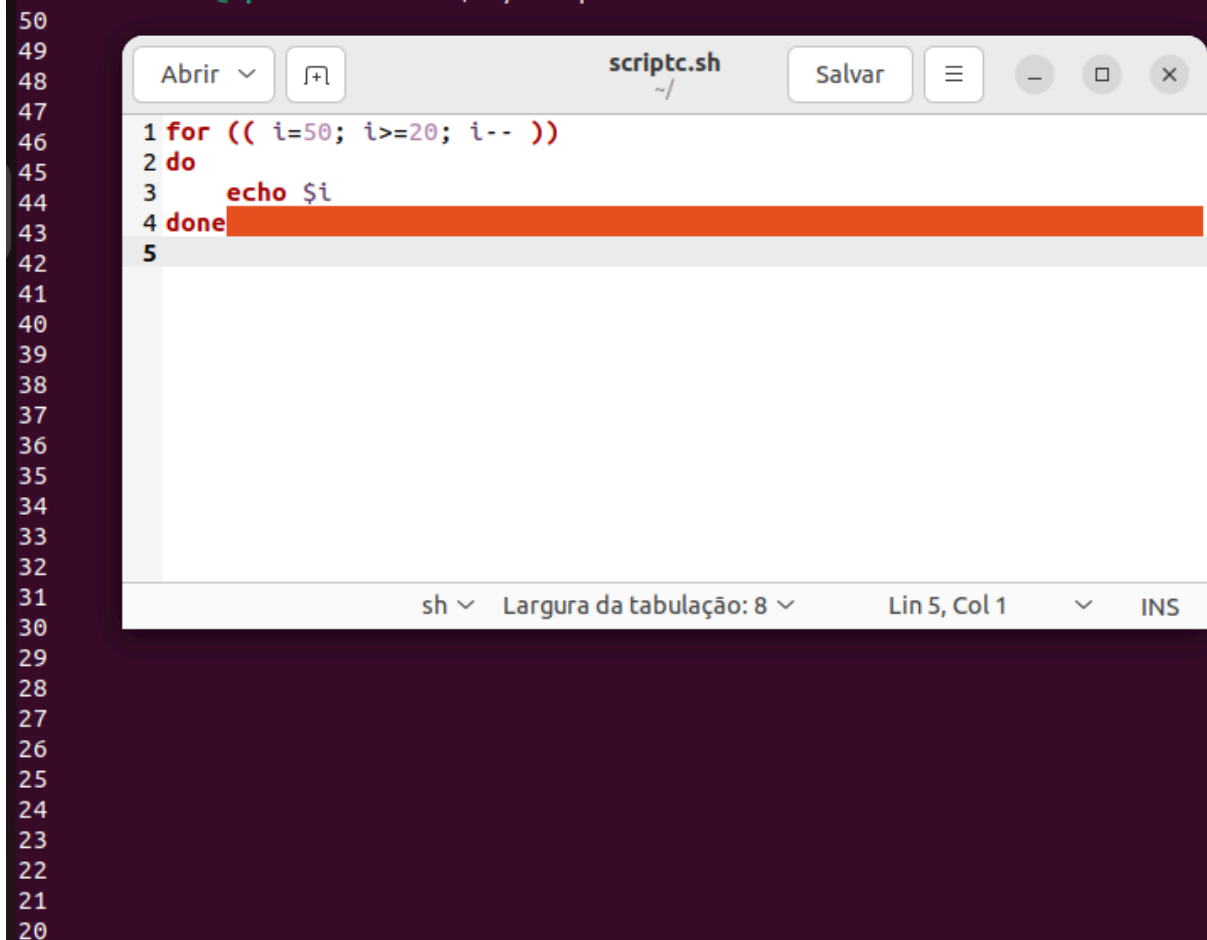
```
2024.1.08.043@OptiPlex-3000:~$ chmod +x scriptnomesifs.sh
2024.1.08.043@OptiPlex-3000:~$ ./scriptnomesifs.sh
Olá, João!
Olá, Maria!
Olá, Carlos!
Olá, Ana!
Olá, Pedro!
```



```
1 nomes="João,Maria,Carlos,Ana,Pedro"
2 IFS=' '
3 for nome in $nomes
4 do
5     echo "Olá, $nome!"
6 done
7 IFS=$' \t\n'
```

5.Crie um script for no estilo C que mostre na tela os números de 50 a 20.

```
2024.1.08.043@OptiPlex-3000:~$ chmod +x scriptc.sh
2024.1.08.043@OptiPlex-3000:~$ ./scriptc.sh
```



```
1 for (( i=50; i>=20; i-- ))
2 do
3     echo $i
4 done
5
```

6.Desenvolva um script que receba um parâmetro e verifique se o valor está entre 0 e 10. Caso sim mostre o triplo do valor. Caso ele esteja entre 10 e 20 mostre o dobro. Caso não esteja nos anteriores apresente uma mensagem.

```
2024.1.08.043@OptiPlex-3000:~$ chmod +x scripttriplo.sh
2024.1.08.043@OptiPlex-3000:~$ ./scripttriplo.sh
./scripttriplo.sh: linha 2: * 3: erro de sintaxe; esperava operando (token de erro é "*" 3)
2024.1.08.043@OptiPlex-3000:~$ ./scripttriplo.sh 4
O triplo de 4 é: 12
2024.1.08.043@OptiPlex-3000:~$ ./scripttriplo.sh 20
O dobro de 20 é: 40
2024.1.08.043@OptiPlex-3000:~$ ./scripttriplo.sh 2
O triplo de 2 é: 6
2024.1.08.043@OptiPlex-3000:~$ ./scripttriplo.sh 10
O triplo de 10 é: 30
2024.1.08.043@OptiPlex-3000:~$
```

```
1 if [[ $1 -ge 0 && $1 -le 10 ]]; then
2     triplo=$(( $1 * 3 ))
3     echo "O triplo de $1 é: $triplo"
4 elif [[ $1 -gt 10 && $1 -le 20 ]]; then
5     dobro=$(( $1 * 2 ))
6     echo "O dobro de $1 é: $dobro"
7 else
8     echo "O número $1 não está nos intervalos especificados (0-10 ou 10-20)."
9 fi
```

7.Explique o que é \$# e faça um script diferente do que foi visto em aula. Faça com dois parâmetros. Use sua criatividade.

\$# é uma variável especial no shell bash que retorna o número de argumentos passados para um script ou função.

```
2024.1.08.043@OptiPlex-3000:~$ ./script$.sh
Por favor, forneça exatamente dois parâmetros.
2024.1.08.043@OptiPlex-3000:~$ ./script$.sh 1 2
Dois parâmetros foram fornecidos!
O primeiro parâmetro é: 1
O segundo parâmetro é: 2
2024.1.08.043@OptiPlex-3000:~$
```

```
1 if [ $# -eq 2 ]; then
2     echo "Dois parâmetros foram fornecidos!"
3     echo "O primeiro parâmetro é: $1"
4     echo "O segundo parâmetro é: $2"
5 else
6     echo "Por favor, forneça exatamente dois parâmetros."
7 fi
```