

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.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ echo "echo Felipe de Godoi Corrêa" > escrevenome
2021.1.08.049@suporte-OptiPlex-3050:~$ echo "[[ -r escrevenome && -x escrevenome ]]" > testecompara
2021.1.08.049@suporte-OptiPlex-3050:~$ bash testecompara && echo "O usuário tem permissão de leitura e execução sobre o script escrevenome" || echo "O usuário não tem permissão de leitura e execução sobre o script escrevenome"
O usuário não tem permissão de leitura e execução sobre o script escrevenome
```

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.”

```
2021.1.08.049@suporte-OptiPlex-3050:~$ nano frutascase.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x frutascase.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ ./frutascase.sh banana
A banana é uma fruta de polpa macia e doce, muito nutritiva e energética. É uma excelente fonte de potássio, vitamina C, vitamina B6 e fibras. Originária do sudeste asiático, é uma das frutas mais consumidas no mundo. Pode ser consumida in natura, usada em sobremesas, saladas de frutas, smoothies, bolos e pães.
```

```
1 |fruta="$1"
2
3 case "$fruta" in
4     uva)
5         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."
6         ;;
7     banana)
8         echo "A banana é uma fruta de polpa macia e doce, muito nutritiva e energética. É uma
excelente fonte de potássio, vitamina C, vitamina B6 e fibras. Originária do sudeste asiático, é
uma das frutas mais consumidas no mundo. Pode ser consumida in natura, usada em sobremesas,
saladas de frutas, smoothies, bolos e pães."
9         ;;
10    morango)
11        echo "O morango é uma fruta vermelha e suculenta, conhecida pelo seu sabor doce e
levemente ácido. É uma excelente fonte de vitamina C, antioxidantes e fibras. Originário da
Europa, é amplamente cultivado em diversas partes do mundo. Pode ser consumido in natura, em
sobremesas, geleias, sorvetes, sucos e saladas."
12        ;;
13    abacaxi)
14        echo "O abacaxi é uma fruta tropical de sabor adocicado e refrescante. Rico em vitamina
C, fibras e fibras, é consumido em diversas formas e é uma das frutas mais consumidas no mundo."
15        ;;
16    *)
17        echo "Fruta não reconhecida."
18    esac
```

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

for é usada para iterar sobre uma lista de itens, while executa um bloco de código repetidamente enquanto uma condição específica for verdadeira, until executa um bloco de código repetidamente enquanto uma condição específica for falsa

```
2021.1.08.049@suporte-OptiPlex-3050:~$ cores=("vermelho" "verde" "azul" "amarelo")
for cor in "${cores[@]"; do
    echo "A cor é: $cor"
done
A cor é: vermelho
A cor é: verde
A cor é: azul
A cor é: amarelo
2021.1.08.049@suporte-OptiPlex-3050:~$ contador=1
while [ $contador -le 5 ]; do
    echo "Contagem: $contador"
    contador=$((contador + 1))
done
Contagem: 1
Contagem: 2
Contagem: 3
Contagem: 4
Contagem: 5
2021.1.08.049@suporte-OptiPlex-3050:~$ arquivo="exemplo.txt"
until [ -e "$arquivo" ]; do
    echo "arquivo ainda não existe."
    sleep 2
done
echo "arquivo $arquivo foi criado!"
arquivo ainda não existe.
arquivo ainda não existe.
```

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

IFS significa "Internal Field Separator" (Separador Interno de Campos) e é uma variável do shell bash que determina como o shell divide palavras em campos quando é realizada a expansão de parâmetros, como a expansão de variáveis e a leitura de entrada.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ dados="João,Silva,30,Analista de Dados"
2021.1.08.049@suporte-OptiPlex-3050:~$ IFS_anterior=$IFS
2021.1.08.049@suporte-OptiPlex-3050:~$ IFS=','
2021.1.08.049@suporte-OptiPlex-3050:~$ read nome sobrenome idade cargo <<< "$dados"
2021.1.08.049@suporte-OptiPlex-3050:~$ IFS=$IFS_anterior
2021.1.08.049@suporte-OptiPlex-3050:~$ echo "Nome: $nome"
echo "Sobrenome: $sobrenome"
echo "Idade: $idade"
echo "Cargo: $cargo"
Nome: João
Sobrenome: Silva
Idade: 30
Cargo: Analista de Dados
2021.1.08.049@suporte-OptiPlex-3050:~$
```

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

```
2021.1.08.049@suporte-OptiPlex-3050:~$ contador=50
while [ $contador -ge 20 ]; do
    echo $contador
    contador=$((contador - 1))
done
50
49
48
47
46
45
44
43
42
41
40
```

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.

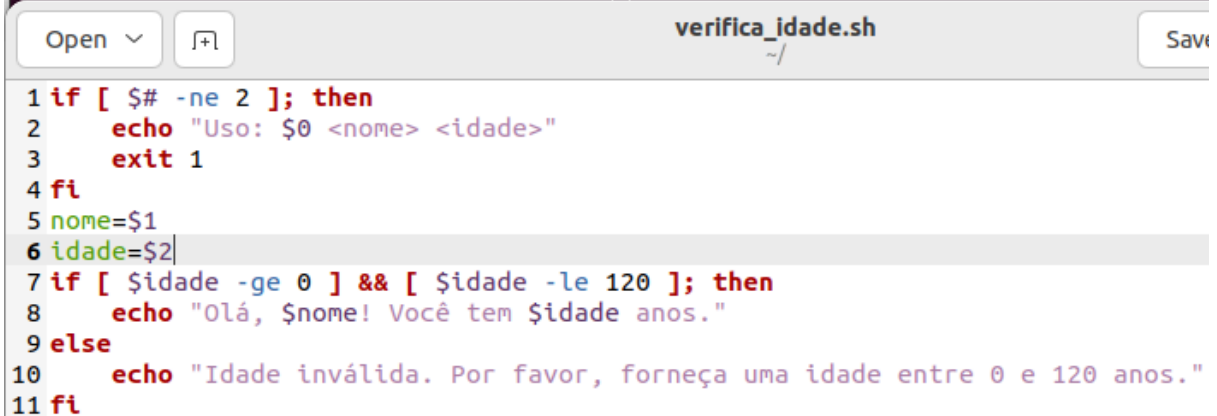
```
2021.1.08.049@suporte-OptiPlex-3050:~$ nano verifica_valor.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x verifica_valor.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ ./verifica_valor.sh 15
O dobro de 15 é 30.
2021.1.08.049@suporte-OptiPlex-3050:~$ ./verifica_valor.sh 35
O valor fornecido não está entre 0 e 20.
2021.1.08.049@suporte-OptiPlex-3050:~$ ./verifica_valor.sh 5
O triplo de 5 é 15.
```

```
Open  ▾  [⌕]  verifica_valor.sh
~/
1 if [ $# -ne 1 ]; then
2     echo "Uso: $0 <valor>"
3     exit 1
4 fi
5 valor=$1
6 if [ $valor -ge 0 ] && [ $valor -le 10 ]; then
7     triplo=$((valor * 3))
8     echo "O triplo de $valor é $triplo."
9 elif [ $valor -gt 10 ] && [ $valor -le 20 ]; then
10    dobro=$((valor * 2))
11    echo "O dobro de $valor é $dobro."
12 else
13    echo "O valor fornecido não está entre 0 e 20."
14 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 do shell script que representa o número de argumentos passados para o script ou função, é o total de parâmetros que foram fornecidos ao script.

```
2021.1.08.049@suporte-OptiPlex-3050:~$ nano verifica_idade.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ chmod +x verifica_idade.sh
2021.1.08.049@suporte-OptiPlex-3050:~$ ./verifica_idade.sh João 30
Olá, João! Você tem 30 anos.
```



```
1 if [ $# -ne 2 ]; then
2     echo "Uso: $0 <nome> <idade>"
3     exit 1
4 fi
5 nome=$1
6 idade=$2
7 if [ $idade -ge 0 ] && [ $idade -le 120 ]; then
8     echo "Olá, $nome! Você tem $idade anos."
9 else
10    echo "Idade inválida. Por favor, forneça uma idade entre 0 e 120 anos."
11 fi
```