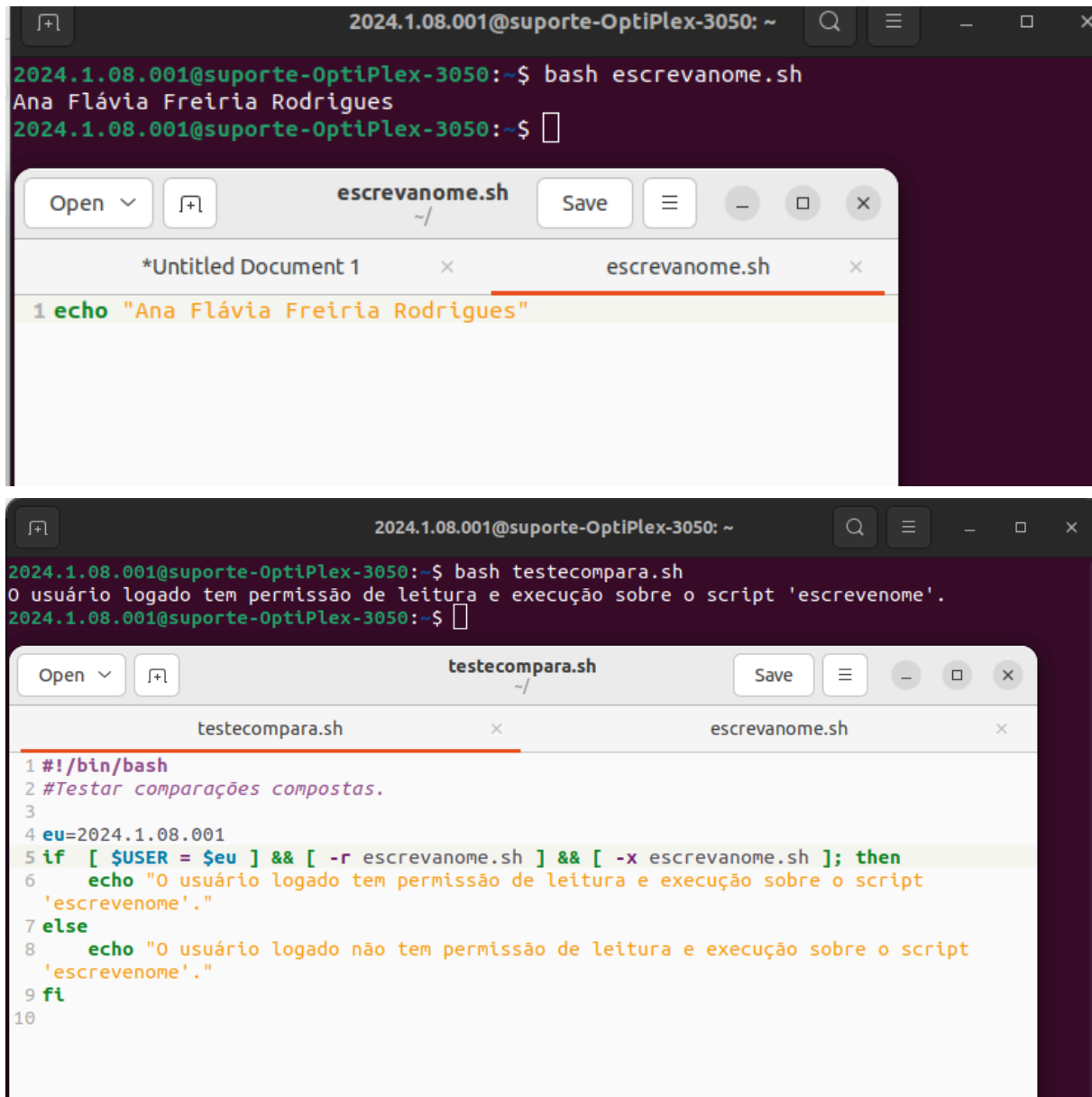


1) Crie um script chamado escrevanome, faça com que a saída desse script seja seu nome completo. Não utilize o comando chmod. Depois crie um script chamado testecompara, utilize o operador AND e verifique se o usuário logado tem permissão r e x sobre o script escrevanome. Mostre o resultado da saída.



The first screenshot shows a terminal window with the command `bash escrevanome.sh` being executed. The output is `Ana Flávia Freiria Rodrigues`. Below the terminal, a code editor window titled `escrevanome.sh` shows the script content: `1 echo "Ana Flávia Freiria Rodrigues"`.

The second screenshot shows a terminal window with the command `bash testecompara.sh` being executed. The output is `0 usuário logado tem permissão de leitura e execução sobre o script 'escrevanome'.`. Below the terminal, a code editor window titled `testecompara.sh` shows the script content:

```
1 #!/bin/bash
2 #Testar comparações compostas.
3
4 eu=2024.1.08.001
5 if [ $USER = $eu ] && [ -r escrevanome.sh ] && [ -x escrevanome.sh ]; then
6     echo "0 usuário logado tem permissão de leitura e execução sobre o script
7     'escrevanome'."
8 else
9     echo "0 usuário logado não tem permissão de leitura e execução sobre o script
10    'escrevanome'."
11 fi
```

2) Crie um script chamado frutascase. Com base no valor da variável fruta mostre uma breve descrição da fruta. Faça com 5 frutas.

```

2024.1.08.001@suporte-OptiPlex-3050:~$ bash frutascase.sh
"As bananas possuem um formato alongado, uma coloração amarela ou vermelha e um sabor doce. Por ser um fruto partenocárpico, a banana não possui sementes, com exceção de uma espécie vendida no mercado indonésio, a Musa balbisiana.
2024.1.08.001@suporte-OptiPlex-3050:~$ bash frutascase.sh
"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.
2024.1.08.001@suporte-OptiPlex-3050:~$ bash frutascase.sh
"O fruto é globoso, com uma profunda depressão no ponto de inserção da haste, que o prende aos ramos. De coloração vermelha ou verde, pode apresentar pequenas manchas esverdeadas ou amareladas. No Brasil, a produção de maçã se concentra em dois cultivares, gala e fuji, que representam em torno de 90% da área plantada.
2024.1.08.001@suporte-OptiPlex-3050:~$ bash frutascase.sh
"O morango é uma fruta vermelha, cuja origem é a Europa. Produzida pelo morangueiro, é um fruto rasteiro. Existem várias espécies de morango, sendo a fragaria a mais comum e cultivada em várias partes do mundo. É uma fruta pouco calórica, apresentando cerca de 32 kcal por 100 gramas de morango.
2024.1.08.001@suporte-OptiPlex-3050:~$ bash frutascase.sh
"A cor da polpa pode variar de amarela a salmão ou róseo-avermelhada. A casca é lisa, verde, tornando-se amarela ou alaranjada quando maduro o fruto.
2024.1.08.001@suporte-OptiPlex-3050:~$

```

```

Open  [icon] frutascase.sh  Save  [icon]  [icon]  [icon]
1 #!/bin/bash
2 fruta=mamao
3 case $fruta in
4 banana)
5 echo "As bananas possuem um formato alongado, uma coloração amarela ou vermelha e um
  sabor doce. Por ser um fruto partenocárpico, a banana não possui sementes, com exceção
  de uma espécie vendida no mercado indonésio, a Musa balbisiana.;;
6 uva)
7 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.;;
8 maçã)
9 echo "O fruto é globoso, com uma profunda depressão no ponto de inserção da haste, que
  o prende aos ramos. De coloração vermelha ou verde, pode apresentar pequenas manchas
  esverdeadas ou amareladas. No Brasil, a produção de maçã se concentra em dois
  cultivares, gala e fuji, que representam em torno de 90% da área plantada.;;
10 morango)
11 echo "O morango é uma fruta vermelha, cuja origem é a Europa. Produzida pelo
  morangueiro, é um fruto rasteiro. Existem várias espécies de morango, sendo a fragaria
  a mais comum e cultivada em várias partes do mundo. É uma fruta pouco calórica,
  apresentando cerca de 32 kcal por 100 gramas de morango.;;
12 mamao)
13 echo "A cor da polpa pode variar de amarela a salmão ou róseo-avermelhada. A casca é
  lisa, verde, tornando-se amarela ou alaranjada quando maduro o fruto.;;
14
15 esac

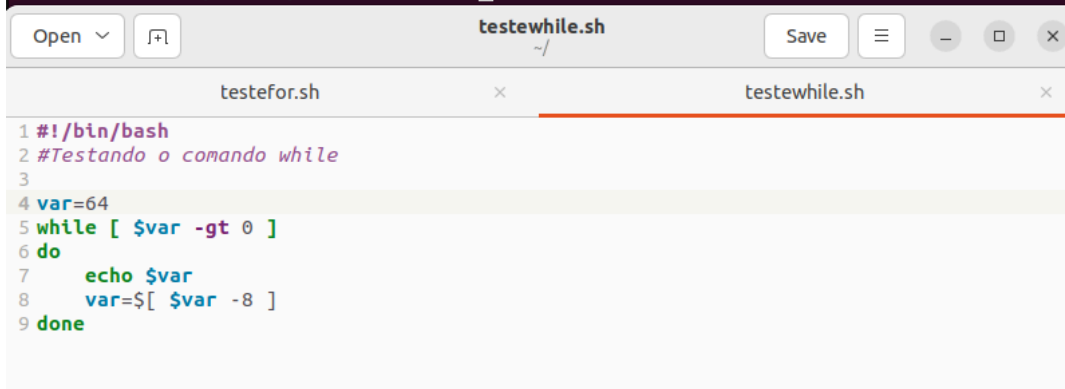
```

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

```
2024.1.08.001@suporte-OptiPlex-3050: ~  
2024.1.08.001@suporte-OptiPlex-3050:~$ bash testeuntil.sh  
Contador: 0  
Contador: 1  
Contador: 2  
Contador: 3  
Contador: 4  
Loop concluído.  
2024.1.08.001@suporte-OptiPlex-3050:~$  
  
testeuntil.sh  
1 #!/bin/bash  
2 # Definindo uma variável de contagem  
3 contador=0  
4  
5 # Utilizando a estrutura until para repetir um bloco de código até que a condição seja verdadeira  
6 until [ $contador -ge 5 ]; do  
7     echo "Contador: $contador"  
8     contador=$((contador + 1)) # Incrementando o contador  
9 done  
10  
11 echo "Loop concluído."  
12
```

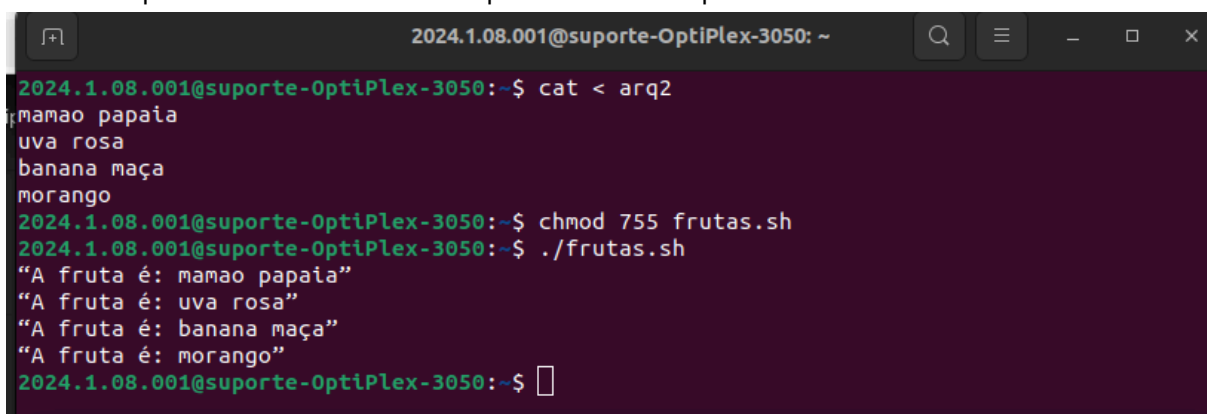
```
2024.1.08.001@suporte-OptiPlex-3050:~$ bash testefor.sh  
Adoro casa  
Adoro viajar  
Adoro dançar  
Adoro lasanha  
Adoro sair  
Adoro cantar  
2024.1.08.001@suporte-OptiPlex-3050:~$  
  
testefor.sh  
1 #!/bin/bash  
2 #Ler valores de uma lista, declarada no próprio comando for  
3  
4 for valor in casa viajar dançar lasanha sair cantar  
5 do  
6 echo Adoro $valor  
7 done
```

```
2024.1.08.001@suporte-OptiPlex-3050:~$ bash testewhile.sh
64
56
48
40
32
24
16
8
2024.1.08.001@suporte-OptiPlex-3050:~$
```

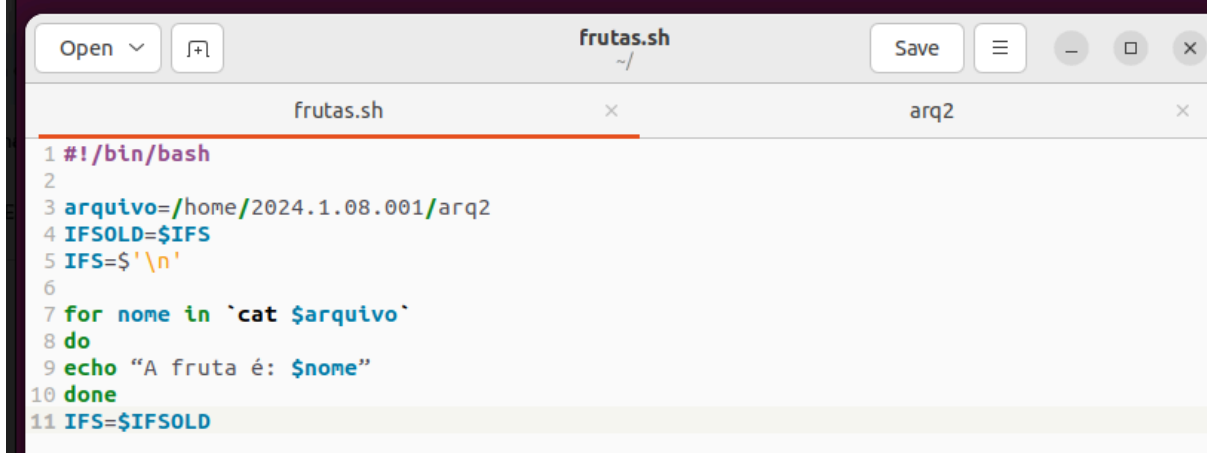


```
1 #!/bin/bash
2 #Testando o comando while
3
4 var=64
5 while [ $var -gt 0 ]
6 do
7     echo $var
8     var=$(( $var - 8 ))
9 done
```

4) Explique o que é IFS e faça um script diferente do que foi visto em aula. Use sua criatividade. São separadores de campo, ou seja, caso os itens iterados sejam compostos por palavras separadas por espaço (como Pão integral), o comando for irá considerar cada palavra como um item separado. Isso é devido à variável de ambiente IFS (Internal Field Separator), a qual define uma lista de caracteres que o shell bash usa como separadores de campos.



```
2024.1.08.001@suporte-OptiPlex-3050:~$ cat < arq2
mamao papaia
uva rosa
banana maça
morango
2024.1.08.001@suporte-OptiPlex-3050:~$ chmod 755 frutas.sh
2024.1.08.001@suporte-OptiPlex-3050:~$ ./frutas.sh
"A fruta é: mamao papaia"
"A fruta é: uva rosa"
"A fruta é: banana maça"
"A fruta é: morango"
2024.1.08.001@suporte-OptiPlex-3050:~$
```



```
1 #!/bin/bash
2
3 arquivo=/home/2024.1.08.001/arq2
4 IFSOLD=$IFS
5 IFS=$'\n'
6
7 for nome in `cat $arquivo`
8 do
9     echo "A fruta é: $nome"
10 done
11 IFS=$IFSOLD
```

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

```
2024.1.08.001@suporte-OptiPlex-3050: ~
2024.1.08.001@suporte-OptiPlex-3050:~$ chmod 755 num.sh
2024.1.08.001@suporte-OptiPlex-3050:~$ ./num.sh
"Número: 50"
"Número: 49"
"Número: 48"
"Número: 47"
"Número: 46"
"Número: 45"
"Número: 44"
"Número: 43"
"Número: 42"
"Número: 41"
"Número: 40"
"Número: 39"
"Número: 38"
"Número: 37"
"Número: 36"
"Número: 35"
"Número: 34"
"Número: 33"
"Número: 32"
"Número: 31"
"Número: 30"
"Número: 29"
"Número: 28"
"Número: 27"
"Número: 26"
"Número: 25"
"Número: 24"
"Número: 23"
"Número: 22"
"Número: 21"
"Número: 20"
2024.1.08.001@suporte-OptiPlex-3050:~$
```

```
Open  *num.sh  Save
frutas.sh  x  arq2  x  *num.sh  x
1 #!/bin/bash
2
3 for (( i = 50 ; i >= 20; i-- ))
4 do
5 echo "Número: $i"
6 done
```

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.

The image shows a code editor window with a file named `ai.ai.sh` open. The script is a bash script that takes a single argument and performs calculations based on its value. Below the editor, a terminal window shows the execution of the script.

```
1 #!/bin/bash
2
3 if [ -z "$1" ]; then
4     echo "Digite um número:"
5     exit 1
6 fi
7 # Transforma em inteiro
8 valor=$1
9 # Verifica se o valor está entre 0 e 10
10 if [ "$valor" -ge 0 ] && [ "$valor" -le 10 ]; then
11     triplo=$((valor * 3))
12     echo "O triplo de $valor é $triplo."
13
14 # Verifica se o valor está entre 10 e 20
15 elif [ "$valor" -gt 10 ] && [ "$valor" -le 20 ]; then
16     dobro=$((valor * 2))
17     echo "O dobro de $valor é $dobro."
18
19 else
20     echo "O valor $valor não está entre 0 e 20."
21 fi
```

Terminal output:

```
2024.1.08.001@suporte-OptiPlex-3050: ~
2024.1.08.001@suporte-OptiPlex-3050:~$ chmod 755 ai.ai.sh
2024.1.08.001@suporte-OptiPlex-3050:~$ ./ai.ai.sh 2
O triplo de 2 é 6.
2024.1.08.001@suporte-OptiPlex-3050:~$ ./ai.ai.sh 15
O dobro de 15 é 30.
2024.1.08.001@suporte-OptiPlex-3050:~$ ./ai.ai.sh 25
O valor 25 não está entre 0 e 20.
2024.1.08.001@suporte-OptiPlex-3050:~$
```

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

The image shows a code editor window with a file named `par.sh` open. The script is a bash script that takes two arguments and calculates the sum of the quadruple of the first argument and the second argument. Below the editor, a terminal window shows the execution of the script.

```
1 #!/bin/bash
2 #Verificação de número de parâmetros
3 if [ $# -ne 2 ]
4 then
5     echo "Digite ao menos dois valores!"
6 else
7     resultado=$(( $1 * 4 ) + $2 )
8     echo "O quadruplo de $1 somado a $2 é $resultado"
9 fi
```

Terminal output:

```
2024.1.08.001@suporte-OptiPlex-3050: ~
2024.1.08.001@suporte-OptiPlex-3050:~$ chmod 755 par.sh
2024.1.08.001@suporte-OptiPlex-3050:~$ ./par.sh
"Digite ao menos dois valores!"
2024.1.08.001@suporte-OptiPlex-3050:~$ ./par.sh 1 2
"O quadruplo de 1 somado a 2 é 6"
2024.1.08.001@suporte-OptiPlex-3050:~$ ./par.sh 5 10
"O quadruplo de 5 somado a 10 é 30"
2024.1.08.001@suporte-OptiPlex-3050:~$
```