1)

a)

```
1   function imprime_msg()
2   {
3   echo "A minha primeira funcao"
4   return 0
5   }
```

```
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO$ ./aula04/aula04e01.sh
A minha primeira funcao
```

b)

A variável $(date) dá-nos o valor do dia e hora do momento, podemos obter apenas a data a usar $(date "+%d/%m/%Y"). Para o nome do computador existe a variável chamada $(hostname) e para o nome do utilizador $(whoami)

c)

```
#!/bin/bash
. ./aula04/aula04e01.sh

imprime_msg
```

```
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO$ ./aula04/aula04e01c.sh
A minha primeira funcao
A data de hoje é: 09/10/2025
O nome do computador é: tiago-asus-tuf-gaming
O nome do utilizador é: tiago
A minha primeira funcao
```

d)

```
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO$ dw()
> {
> date
> who
> }
```

Esta função executa sempre, independentemente do diretório atual

2)

a) Esta função tem como argumento a entrada na linha de execução do ficheiro (valor numérico), ela vai dar match no argumento e se for igual a 1, 2 ou 3 da echo no nome (em extenso) do número, qualquer outro escreve "Outro número"

b)

```bash
#!/bin/bash

function numeric_to_string() {
    case "$1" in
        1)
            echo "Um"
            return 1
            ;;
        2)
            echo "Dois"
            return 2
            ;;
        3)
            echo "Três"
            return 3
            ;;
        *)
            echo "Outro número"
            return 0
            ;;
    esac
}

# Chamada da função
numeric_to_string $1

valor_retorno=$?
echo "O valor de retorno foi: $valor_retorno"
```

c)

```
ret ()
{
    return $1
}
```

```
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO$ ret 1; echo $?
1
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO$ ret 255; echo $?
255
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO$ ret 256; echo $?
0
```

A gama de valor de retorno vai até 255 (4bytes)

d) e e)

```
#!/bin/bash

function comparator() {
    if [[ "$1" -eq "$2" ]]; then
        return 0
    else
        if [[ "$1" -gt "$2" ]]; then
            return 1
        else
            return 2
        fi
    fi
}

echo "Digite dois números"
read num1 num2
comparator $num1 $num2

#comparator "$1" "$2" #Com os argumentos da linha de comando

valor_retorno=$?

if [[ $valor_retorno -eq 0 ]]; then
    echo "Números iguais"
else
    if [[ $valor_retorno -eq 1 ]]; then
        echo "$num1 é maior"
    else
        echo "$num2 é maior"
    fi
fi
```

3.

a) O código começa por criar uma variável chamada lista, que contém um array de números de 1 até 10, a expressão {1..10}, já antes analisada, significa o intervalo de números de 1 até 10. Depois o script inicializa um loop for para iterar por todos os elementos da lista (${lista[@]}) e guarda-os na variável i, e a cada iteração vai dar print de i.

```
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO$ chmod u+x aula04/aula04e03a.sh
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO$ cd aula04
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$ ./aula04e03a.sh
1
2
3
4
5
6
7
8
9
10
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$
```

b)

Primeiro o programa vai criar uma variável chamada lista, que tem um array $(seq 2 3 15), isto significa que vai guardar todos os números de 3 em 3 desde 2 até 15, ou seja 2, 5, 8, 11, 14.

echo $(lista[@]) = 2 5 11 14 (elementos da lista)

echo $(#lista[@]) = 5 (quantidade de elementos da lista)

echo $(!lista[@]) = 0 1 2 3 4 (indexs dos elementos da lista)

Depois o script começa um ciclo for, que vai desde i = 0 até i < $lista[@] (que é igual ao comprimento da lista), e vai substituir o valor do index i da lista pelo seu valor + 1, logo a lista agora é 3, 6, 9, 12, 15; e no final dá print ao valor novo de indíce i.

unset lista[1] -> elimina o elemento de índice 1 da lista (neste caso = 6)

unset lista[3] -> elimina o elemento de índice 3 da lista (neste caso = 12)

Agora a lista tem cardinalidade de apenas 3.

No final cria outro ciclo for para iterar por cada índice da lista (!lista[@]), e faz echo de cada valor de índice i da lista.

```
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$ chmod u+x aula04e03b.sh
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$ ./aula04e03b.sh
vals 2 5 8 11 14
count 5
index 0 1 2 3 4
0: 3
1: 6
2: 9
3: 12
4: 15
count 3
0: 3
2: 9
4: 15
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$
```

c)

Primeiro, o script vai criar um "Associative array" designado por -A chamado assArray, depois vai colocar na array valores, neste formato assArray[key] = valor, aqui a key funciona como se fosse um índex.

assArray[Aveiro]=10 (Cria a chave Aveiro com valor = 10)

assArray[Porto]=NA (Cria a chave Porto com valor = NA)

assArray+=([Lisboa]=5) (Adiciona à chave Lisboa o valor = 5, como não existia ainda esta chave, o script cria uma)

Depois vai imprimir os valores da array echo vals ${assarray[@]},

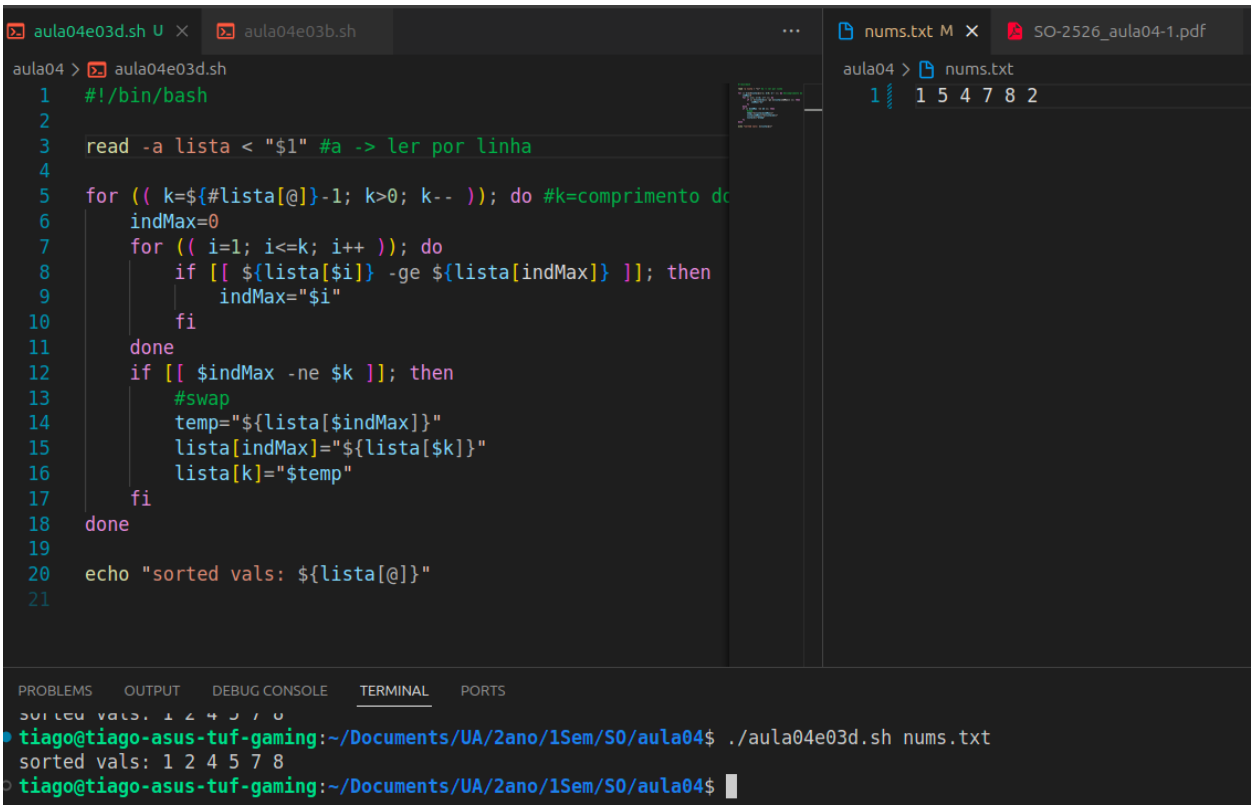
Vai imprimir o valor do comprimento da array = 3 echo count ${$assArray[@]}

Vai imprimir os índices, ou, neste caso, as chaves do array echo index ${!assArray[@]}

No final imprime o valor associado à chave de Lisboa (imprime 5) echo val Lisboa ${assArray[Lisboa]}

```
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$ chmod u+x aula04e03c.sh
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$ ./aula04e03c.sh
vals NA 5 10
count 3
index Porto Lisboa Aveiro
val Lisboa 5
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$
```

d)


aula04e03d.sh U
aula04e03b.sh
aula04 > aula04e03d.sh
#!/bin/bash
read -a lista < "$1" #a -> ler por linha
for ((k=${#lista[@]}-1; k>0; k--)); do #k=comprimento d
indMax=0
for ((i=1; i<=k; i++)); do
if [[${lista[$i]} -ge ${lista[indMax]}]]; then
indMax="$i"
fi
done
if [[$indMax -ne $k]]; then
#swap
temp="${lista[$indMax]}"
lista[indMax]="${lista[$k]}"
lista[k]="$temp"
fi
done
echo "sorted vals: ${lista[@]}"
nums.txt M
SO-2526_aula04-1.pdf
aula04 > nums.txt
1 5 4 7 8 2
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$ ./aula04e03d.sh nums.txt
sorted vals: 1 2 4 5 7 8
tiago@tiago-asus-tuf-gaming:~/Documents/UA/2ano/1Sem/SO/aula04$