

**1. Quina és la sortida d'aquesta comanda?**

```
for i in "Hola Mon Bash";  
do  
echo $i  
done
```

- a. "Hola Mon Bash"
- b. Hola  
Mon  
Bash
- c. Hola Mon Bash

**2. Què passa en una iteració for si itera sobre un string buit?**

- a. Dona error
- b. Itera una vegada sobre el null i surt del loop
- c. No executa el codi de dintre i passa al següent

**3. Què passarà en aquesta comanda si la variable num equival a 5?**

```
num=5  
if [ $num -eq 5 ]; then  
echo "$num equival a 5"  
else  
echo "$num no equival a 5"
```

- a. Imprimirà "5 equival a 5"
- b. Imprimirà no equival a 5
- c. Té un error de sintaxi

**4. Quina comanda imprimirà els fitxers que comencen per hosts i qualsevol strings?**

- a. `grep -e "^(hosts)."`
- b. `grep -e "[hosts]."`
- c. `ls hosts*`

5. Fixa't en el fitxers i després en l'expressió regular. Quines creus que imprimirà el grep?

```
1  Loopback:
2  127.0.0.1
3  Hosts:
4  192.168.122.220
5  Network:
6  192.168.122
7  Default gateway:
8  192.168.122.1
9  DNS:
10 10.100.1.252
11 10.100.1.210
```

```
grep -e "(\d{1,3}\.){3}\."
```

- a. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- b. 2, 4, 8, 10 i 11
- c. 2, 4, 6, 8, 10, 11
- d. 10 i 11

6. Quina és la funció del parametre \$?

```
echo $?
0
```

- a. Per llistar el nombre de paràmetres usats
- b. Per llistar el codi d'error de l'ultima , i en aquest cas s'ha executat correctament
- c. Per llista el codi d'error d el'ultima comanda executada, i en aquesta cas s'ha executat amb errors,
- d. Mostra l'argument numero X

7. Com fem per escapar els caràcters \$, ?, #, =, "",

- a. No es poden escapar aquesta caràcters
- b. Es poden escapar amb /
- c. Es poden escapar amb \

### 8. S'executarà aquesta comanda?

```
for i in (1..10)
do
    echo "$i"
done
```

1. No, li falta la ; a l'expressió
2. Només s'executarà la primera i l'última opció
3. S'executarà sense problemes

### 9. Quin output imprimeix aquest codi?

```
i=0
until [[ $i -ge 3 ]]; then
    echo $i
    i=$((i+1))
done
```

- a. 0  
1  
2
- b. 0  
1  
2  
3
- c. 0 1 2 3
- d. Res, la comanda hauria de ser while

### 10. Per a que serveix aquest codi?

```
if [[ -z $USUARI ]]; then
    echo "s'ha de proporcionar un nom d'usuari"
else
    echo "hola $USUARI"
fi
```

- a. Comprova que el fitxer existeixi, si no, imprimeix "s'ha de proporcionar un nom d'usuari"
- b. Comprova que el directori existeixi, si no, imprimeix "s'ha de proporcionar un nom d'usuari"
- c. Comprova que la cadena existeixi, si no, imprimeix "s'ha de proporcionar un nom d'usuari"
- d. Comprova que l'última comanda s'hagi executat bé, si no, imprimeix "s'ha de proporcionar un nom d'usuari"

### 11. Què passa amb aquesta comanda?

```
for i in (seq 1 10);  
do  
    echo "nombre: $i"  
done
```

- a. Té un error de sintaxi i no s'executarà
- b. Res, està tot bé i s'executarà fent el loop
- c. S'executarà però no imprimirà res