

# Capítulo 4

## ===Exercício 1===

Sobre a variável `BASH_REMATCH`, podemos afirmar que:

- I. É um vetor;
- II. É alimentada pelo comando `test` usando Expressões Regulares;
- III. Guarda o código de erro da última instrução;
- IV. No seu índice 0 (zero) tem todo o texto casado pela Expressão Regular;
- V. Nos índices superiores a zero (0) tem os textos que foram casados com os grupos da Expressão Regular;
- VI. Esta variável tem o ponteiro para o próximo comando `test`.

Estão certas as linhas:

- A) Todas estão certas
- B) II, III e VI estão certas
- C) II, III, V e VI
- D) I, II, IV e V
- E) Todas estão erradas

A resposta correta é a D

## ===Exercício 2===

Fazer um programa que imprima a hora do sistema (horas e minutos) no formato 12 horas (ex.: 7:00 am/pm).

Observações:

- 1. 12:00 e 00:00 não são AM nem PM
- 2. A hora do sistema pode ser obtida com o comando `date +%H` e os minutos com `date +%M`

```
$ cat exerc4-2.sh
#!/bin/bash
# Exercício 2 do capítulo 4
# Passa a data do sistema para o formato AM - PM

#Hora=$(date +%H)
#Min=$(date +%M)
Hora=$1
Min=$2
Horario=$Hora:$Min
AP=AM
```

```
[[ $Horario =~ (00|12):00 ]] && {
    echo $Horario
    exit
}
((10#$Hora >= 12)) && {      # Especificando que a base é 10 para não
dar erro às 08 e 09 horas
    let Hora-=12
    AP=PM
}
echo $Hora:$Min $AP
```

### ===Exercício 3===

Escreva um programa que dê bom dia, boa tarde ou boa noite, de acordo com a hora apresentada pelo comando date.

```
$ cat exerc4-3.sh
#!/bin/bash
# Exercício 3 do capítulo 4
# Dá bom dia, boa tarde ou boa noite dependendo
#+ do horário do sistema
Hora=$(date +%H)
case $Hora in
    # Poderia ser case $(date +%H) in
    0[0-9]|1[01]) echo Bom dia
                ;;
    1[2-7]) echo Boa tarde
            ;;
    *) echo Boa noite
        ;; # Este terminador (;;) antes do esac é opcional
esac
```

### ===Exercício 4===

Gotas de chuva. Se o parâmetro recebido for:

Múltiplo de	Escreva
3	Ping
5	Pang
7	Pong

Se não for múltiplo de nenhum desses, escreva o número. Ex:

```
$ chuva.sh 13
13
$ chuva.sh 9
Ping
$ chuva.sh 21
PingPong
$ chuva.sh 105
PingPangPong
$ cat exerc4-4.sh
```

```
#!/bin/bash
if [[ $# -ne 1 ]]
then
    echo "Uso: $0 <NÚMERO>" >&2
    exit 1
fi
Saida=
(($1 % 3 == 0)) && Saida+=Ping      # Poderia ser (($1 % 3)) || Saida+=Ping
(($1 % 5 == 0)) && Saida+=Pang     # Poderia ser (($1 % 5)) || Saida+=Pang
(($1 % 7 == 0)) && Saida+=Pong     # Poderia ser (($1 % 7)) || Saida+=Pong
[[ -z $Saida ]] && echo $1 || echo $Saida      # Saida está vazia?
```

## ===Exercício 5===

5. Seu *script* receberá um ano e você dirá se ele é bissexto ou não.

Um ano para ser bissexto tem de atender às três seguintes exigências:

1. Tem de ser divisível por 4;
2. Exceto os que são divisíveis por 100;
3. A não ser que seja divisível por 400.

Por exemplo:

1950 não foi bissexto por não ser divisível por 4 (um macete matemático: para ser divisível por 4 basta que o número formado pelos 2 últimos algarismos do ano – nesse caso 50 – seja divisível por 4);

1900 não foi bissexto porque era múltiplo de 100;

2000 foi bissexto porque apesar de ser divisível por 100 (como 1900 que acabamos de ver), também o é por 400.

**\$ cat exerc4-5.sh**

```
#!/bin/bash
if [[ $# -ne 1 ]]                # Testa se recebeu 1 parâmetro
then
    echo "Uso: $0 " >&2          # Manda mensagem para stderr
    exit 1
fi

[[ $1 =~ [0-9]{4} ]] || {      # O parâmetro tem 4 algarismos?
    echo "Uso: $0 " >&2          # Manda mensagem para stderr
    exit 1
}

if (($1 % 400 == 0))            # Se for múltiplo de 400, tb será de 4
then
    echo bissexto
else
    if (($1 % 4 == 0) && (($1 % 100 != 0)) # Múltiplo de 4 e não de 100?
    then
        echo bissexto
```

```

else
    echo não bissexto
fi
fi

```

## ===Exercício 6===

Sobre os terminadores do comando **case** podemos dizer que:

	<i>;;</i>	<i> ; ?</i>	<i> ; ; ?</i>
<b>A</b>	Sempre executa o próximo bloco	Termina a execução do case	Testa as próximas condições até sair
<b>B</b>	Termina a execução do case	Testa as próximas condições até sair	Sempre executa o próximo bloco
<b>C</b>	Testa as próximas condições até sair	Sempre executa o próximo bloco	Termina a execução do case
<b>D</b>	Sempre executa o próximo bloco	Testa as próximas condições até sair	Termina a execução do case
<b>E</b>	Termina a execução do case	Sempre executa o próximo bloco	Testa as próximas condições até sair

Qual alternativa na tabela acima está correta?

- A) Alternativa A
- B) Alternativa B
- C) Alternativa C
- D) Alternativa D
- E) Alternativa E

Resposta correta é a alternativa **E**