

## Exercícios 01

Escreva um programa em C para:

**1.1** Ler um inteiro N. A seguir ler N caracteres e imprimir para cada um a palavra “Sim” se o caractere lido é uma letra maiúscula e “Não” caso contrário.

**1.2** Escreva um algoritmo para ler um inteiro N. A seguir ler N caracteres. Para cada caractere lido escrever o seu respectivo código ASCII.

**1.3** Ler uma quantidade indeterminada de caracteres. Conforme o caractere lido o programa deve executar uma das seguintes ações:

**Letra 'M'** (maiúscula): Exibir uma linha com todas as letras maiúsculas (A-Z) separadas por um espaço.

**Letra 'm'** (minúscula): Exibir uma linha com todas as letras minúsculas (a-z) separadas por um espaço.

**Letra 'D' ou 'd'**: Exibir uma linha com todos os dígitos (0-9) separados por um espaço.

**Letra 'F'** (Maiúscula): O programa deve ser finalizado exibindo quantos caracteres foram digitados (incluindo a letra 'F').

**Qualquer outro caractere:** Exibir uma sequência de 10 caracteres '@'.

OBS: As linhas com as letras ou dígitos devem ser impressas com o uso de repetição.

**1.4** Ler uma quantidade indeterminada de trincas de caracteres. O primeiro caractere da trinca deve ser validado através de repetição para aceitar apenas as letras 'C', 'D' ou 'F' (maiúsculas). Escrever os dois últimos caracteres da trinca na ordem indicada pelo primeiro caractere como especificado a seguir:

'C': ordem crescente

'D': ordem decrescente.

O programa termina quando o primeiro caractere digitado for 'F' (neste caso sem ler os demais).

**1.5** Ler um conjunto indeterminado de caracteres. Para cada caractere digitado escrever uma mensagem indicando se ele é uma letra **maiúscula**, **minúscula**, **dígito** ou **sem classificação**. O programa termina ao ser informada a letra 'f' (independente de ser maiúscula ou minúscula) imprimindo a sua classificação.

**1.6** Ler um inteiro N. A seguir ler N caracteres representando cada um deles um dígito hexadecimal (considere que ao ser informada uma letra ela será maiúscula). Para cada caractere informado escrever o seu respectivo valor em decimal. Considere que serão informados apenas caracteres válidos.

**1.7** Ler um inteiro N. A seguir ler N duplas de caracteres onde cada um representa um dígito hexadecimal. Validar a leitura de cada caractere separadamente para aceitar apenas dígitos hexadecimais (considere que ao ser informada uma letra ela será maiúscula). Caso seja informado um caractere inválido escrever a mensagem “Dígito hexadecimal inválido” e repetir a leitura até que seja válido. Escrever para cada dupla o resultado da soma dos dois dígitos na base decimal.

**1.8** Ler uma quantidade indeterminada de caracteres. Caso seja informado uma letra maiúscula escrever as letras anterior e seguinte (considerando a ordem alfabética) da letra informada. Considere que a letra anterior a letra **A** é a letra **Z** e que a seguinte a letra **Z** é a letra **A**. Para qualquer outro caractere informado o programa deve ser encerrado.

**1.9** Ler um inteiro T que representa a quantidade de letras 'X' que serão impressas. A seguir imprimir uma linha com T letras 'X'.

**1.10** Imprimir uma linha com 5 asteriscos. A seguir ler uma quantidade indeterminada de caracteres. O programa termina ao ser informado o caractere '@'. Conforme o caractere lido o programa deve executar uma das seguintes ações:

**Caractere '+'**: Escrever uma linha de asteriscos com um caractere a mais que a impressa anteriormente.

**Caractere '-'**: Escrever uma linha de asteriscos com um caractere a menos que a impressa anteriormente. Caso a linha impressa anteriormente possua apenas um asterisco ela deve ser impressa novamente com um único asterisco.

**Caractere '@'**: Termina o programa imprimindo a quantidade de vezes que foi informado o caractere '+' e a quantidade de vezes que foi informado o caractere '-'.

**Qualquer outro caractere:** Nenhuma ação será executada.

**1.11** Ler inteiro T e um caractere. A seguir imprimir T linhas com o caractere conforme o exemplo abaixo. Para um T igual a 3 e o caractere 'X' a saída será:

```
X
XX
XXX
```

## Dados para testar os programas:

### 1.1)

[Entrada]	[Saída]
5	
a	Não
A	Sim
1	Não
@	Não
Z	Sim

### 1.2)

[Entrada]	[Saída]
4	
A	65
a	97
#	35
2	50

### 1.3)

[Entrada]	[Saída]
M	A B C D E F G H I J K L M N O P Q R S T U V X Y Z
m	a b c d e f g h i j k l m n o p q r s t u v x y z
D	0 1 2 3 4 5 6 7 8 9
\$	@@@@@@@@@@@
d	0 1 2 3 4 5 6 7 8 9
F	6

### 1.4)

[Entrada]	[Saída]
A	
C s z	s z
B	
C z s	s z
C a B	B a
D s z	z s
D z s	z s
F	

### 1.5)

[Entrada]	[Saída]
a	Minúscula
B	Maiúscula
3	Dígito
!	Sem classificação
f	Minúscula

### 1.6)

[Entrada]	[Saída]
5	
1	1
A	10
E	14
5	5
F	15

### 1.7)

[Entrada]	[Saída]
5	
W	Dígito hexadecimal inválido
!	Dígito hexadecimal inválido
A R	Dígito hexadecimal inválido
1	11
3 5	8
A B	21
A 0	10
F F	30

### 1.8)

[Entrada]	[Saída]
C	B D
F	E G
X	W Y
Z	Y A
W	V X
A	Z B
!	

### 1.9)

[Entrada]	[Saída]
5	XXXXX

### 1.10)

[Entrada]	[Saída]
	*****
-	****
-	***
-	**
+	***
A	
+	****
@	Quantidade de +: 2
	Quantidade de -: 3

### 1.11)

[Entrada]	[Saída]
4 A	
	A
	AA
	AAA
	AAAA