

Lista 14 - Caracteres, String e Structs

Caracteres:

14.1 Escreva um algoritmo para ler 5 caracteres e imprimir para cada um a palavra **“Sim”** se é uma letra maiúscula e **“Não”** caso contrário.

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

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

[Entrada]	[Saída]
2 (N)	
a	97
c	99

14.3 Escreva um algoritmo para ler várias duplas de caracteres e escreva uma mensagem indicando se cada dupla está em ordem crescente ou decrescente. O programa termina ao ser informado 2 caracteres iguais.

[Entrada]	[Saída]
a b	Crescente
a A	Decrescente
z z	

14.4 Escreva um algoritmo para ler várias trincas de caracteres. O primeiro caractere deve ser validado através de repetição para aceitar apenas as letras 'C','D' ou 'F'. Escrever os dois últimos caracteres da trinca na ordem indicada pelo primeiro caractere como especificado a seguir. 'C' : ordem crescente e 'D' : ordem decrescente. O programa termina quando o primeiro caractere digitado for 'F' (neste caso sem ler os demais).

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

14.5 Escreva um algoritmo para ler vários caracteres. Para cada caractere lido faça o seguinte: Se for uma letra minúscula escreva a sua correspondente em maiúscula. Se for uma letra maiúscula escreva a mensagem **“Letra maiúscula”**. Se for um caractere que não é uma letra escrever **“Não é letra”**. O programa termina ao ser informada o caractere '@' imprimindo a sua classificação.

[Entrada]	[Saída]
a	A
B	Letra maiúscula
3	Não é letra
@	Não é letra

14.6 Escreva um algoritmo para ler vários caracteres. Para cada caractere lido escrever uma mensagem indicando se ele é uma letra maiúscula, minúscula, dígito ou sem classificação. O programa termina ao ser informada a letra 'f' imprimindo a sua classificação.

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

14.7 Reescreva o exercício **14.3** fazendo com que cada caractere lido seja validado através de repetição para aceitar apenas letras (maiúsculas ou minúsculas). O programa deve considerar uma letra minúscula igual a sua respectiva maiúscula.

[Entrada]	[Saída]
1	
Z	
!	
a	Decrescente
a B	Crescente
B a	Decrescente
z Z	

14.8 Escreva um algoritmo para ler um inteiro T que representa a quantidade de letras 'X' que serão impressas. A seguir imprimir uma linha com T letras 'X'.

```
[Entrada] [Saída]
5         XXXXX
```

14.9 Escreva um algoritmo para ler inteiro T e dois caracteres. A seguir imprimir uma linha com T caracteres alternando a impressão dos caracteres lidos conforme o exemplo abaixo.

```
[Entrada] [Saída]
9  x  -    x-x-x-x-x
```

14.10 Escreva um algoritmo para ler inteiro T e um caractere. A seguir imprimir T linhas com o caractere conforme o exemplo abaixo.

```
[Entrada] [Saída]
4  x      x
          xx
          xxx
          xxxx
```

14.11 Escreva um algoritmo para ler inteiro T e dois caracteres. A seguir imprimir T linhas com os caracteres conforme o exemplo abaixo.

```
[Entrada] [Saída]
4  x  -    x-
          xx--
          xxx---
          xxxx----
```

14.12 Escreva um algoritmo para ler inteiro T e dois caracteres. A seguir imprimir T linhas com os caracteres conforme o exemplo abaixo.

```
[Entrada] [Saída]
4  x  -    ----x----
          ---xxx---
          --xxxxx--
          -xxxxxxx-
```

Strings:

14.13 Ler um valor N. A seguir ler N palavras. Para cada palavra lida escrever a sua respectiva quantidade de caracteres.

```
[Entrada] [Saída]
2
tsi        3
vermelho   8
```

14.14 Ler um valor N. A seguir ler N palavras. Para cada palavra lida escrever a primeira e a última letra da palavra.

```
[Entrada] [Saída]
2
tsi        t  i
informatica i  a
```

14.15 Ler uma quantidade indeterminada de frases. Para cada frase imprima a quantidade de espaços em branco que ela contém. O programa se encerra quando for digitada uma frase que não contém espaços em branco.

```
[Entrada] [Saída]
Isto e um teste      3
Curso de C           2
TSI
```

14.16 Ler um valor N. A seguir ler N palavras. Escrever cada palavra lida ao contrário.

```
[Entrada] [Saída]
2
BRASIL      LISARB
PELOTAS     SATOLEP
```

14.17 Ler várias duplas de dados representando uma letra e uma palavra. Para cada dupla escrever quantas vezes a letra aparece na palavra. O programa termina ao ser digitada a letra 'f' (nesta situação a palavra não deve ser lida).

```
[Entrada] [Saída]
```

```

a amora      2
a pessego    0
f

```

14.18 Ler um valor N. A seguir ler N palavras. Para cada palavra lida trocar todas as letras 'A' pelo caractere '-'. Após a troca escrever a palavra modificada.

```

[Entrada]      [Saída]
2
amora          -mor-
informatica    inform-tic-

```

14.19 Ler um valor N. A seguir ler N duplas de palavras. Escrever para cada dupla: IGUAIS, CRESCENTE ou DECRESCENTE conforme as palavras da dupla.

```

[Entrada]      [Saída]
3
tsi tsi        IGUAIS
uva amora      DECRESCENTE
gremio inter   CRESCENTE

```

14.20 Ler uma quantidade indeterminada de trincas de dados representando respectivamente um código (1.Crescente 2.Decrescente) e duas palavras. Para cada trinca escrever a dupla de palavras em ordem crescente ou decrescente, separadas por vírgula, conforme o código digitado. O programa termina ao ser informado um código inválido. Nesta situação as duas palavras não serão lidas.

```

[Entrada]      [Saída]
1 (crescente)  Beatriz Ana   Ana,Beatriz
2 (decrescente) carla Ze     carla,Ze
4

```

14.21 Ler várias palavras. Escrever "OK" se a string armazena apenas letras minúsculas, caso contrário escrever "NAO". O programa termina ao ser informada a palavra "fim" que não deve ser processada.

```

[Entrada]      [Saída]
teste         SIM
tSi           NAO
fim

```

14.22 Ler várias palavras (considere que serão informadas apenas palavras em letras minúsculas). Escrever o número de vogais existentes em cada palavra. O programa se encerrará quando for digitada a palavra "fim".

```

[Entrada]      [Saída]
informatica    5
tsi            1
fim

```

14.23 Ler uma letra e uma palavra. Copiar para outra string a palavra informada retirando todas as letras iguais a informada Escrever a string copiada.

```

[Entrada]      [Saída]
a informatica  informtic

```

14.24 Ler várias duplas de dados representando respectivamente um código (1.antes 2.depois 3.fim) e um endereço de e-mail. Copiar para outra string a parte do e-mail que está antes ou depois do @ (conforme o código informado). Escrever a string copiada. O programa termina ao ser fornecido o código 3.

```

[Entrada]      [Saída]
2 (depois)     fulano@ifsul.edu.br   ifsul.edu.br
1 (ante)       ciclano@hotmail.com   ciclano
3 (fim)

```

14.25 Escreva um programa para ler 8 palavras. Imprimir a menor e a maior palavra considerando a ordem alfabética.

```

[Entrada]      [Saída]
amora
uva

```

Strucs

14.26 Definir um tipo chamado *Horario* para armazenar um horário composto de: hora, minuto, segundo.

14.27 Escreva um programa em C para ler dois horários. Escreva o menor horário no formato HH:MM:SS. No caso de igualdade escrever a mensagem "Horários iguais".

14.28 Definir um tipo chamado *Aluno* para armazenar os seguintes dados de um aluno: nome, número de matrícula e data de nascimento (tipo *Data*).

14.29 Escrever um programa em C para ler a data de hoje armazenando-o em uma variável do tipo *Data*. A seguir ler uma quantidade indeterminada de dados de alunos (*Aluno*). Para cada aluno lido escrever se ele já completou 18 anos até a data informada seguido do seu **nome** e sua **idade** conforme modelo ("**Fulano** é maior de 18 e possui **20** anos."). O programa termina ao ser informado o valor 99 para o dia. Nesta situação o mês e o ano não devem ser lidos.

14.30 Escreva um programa em C para ler os dados de 5 alunos (um vetor do tipo *Aluno*). A seguir ler uma data(*Data*). Copiar para um vetor de *datas* (*Data*) as datas de nascimento dos alunos que nasceram no mês da data informada e um vetor de *nomes* com os índices dos alunos do vetor *Aluno*. Após o término da cópia escrever os vetores de *datas* e o nome dos alunos de acordo com os índices salvos no vetor de *nomes*.

*Entregar 5 exercícios escolhendo pelo menos um de cada tipo (Character, String e Strucs)!