



- 1) Solicite a entrada de um conjunto de valores (separados por vírgula) com o *prompt*. A seguir, mostre a quantidade de valores informados, os valores na ordem em que foram digitados, em ordem crescente e ordem decrescente.
- 2) Construa uma função que receberá duas *Strings* e que retornará *True* ou *False* caso todos os caracteres (independentemente de ser maiúsculo ou minúsculo) estejam contidos em ambas as palavras.

Exemplos:

- “Mar” e “Ram” retornará *True*.
- “marte” e “morte” retornará *False*.
- “Banana” e “Ban” retornará *True*.
- “CaRoLiNa” e “cArOllnA” retornará *True*.
- “Hum” e “H um” retornará *False*.

- 3) Escreva um programa que receba um pequeno texto e a seguir mostre a quantidade de palavras digitadas e a seguir as palavras em ordem alfabética (palavra é definida por qualquer sequência de caracteres delimitada por espaços em branco).
- 4) Escreva um programa que receba um pequeno texto e a seguir mostre as palavras ordenadas pela quantidade de caracteres, da maior para a menor, uma em cada linha. Caso exista mais de uma palavra com o mesmo tamanho, elas deverão ser exibidas na mesma linha, separadas por vírgula.
- 5) Crie um programa que solicita um valor numérico ao usuário e imprima cada um dos seus dígitos por extenso.

Exemplo:

Número informado: **4571**

Resultado: quatro, cinco, sete, um

- 6) Escreva um programa que, a partir de um nome de uma pessoa informado pelo usuário, exiba suas iniciais. As iniciais são formadas pela primeira letra de cada nome, sendo que todas deverão aparecer em maiúsculas na saída do programa. Note que os conectores e, do, da, dos, das, de, di, du não são considerados nomes e, portanto, não devem ser considerados para a obtenção das iniciais. As iniciais devem ser impressas em maiúsculas, ainda que o nome seja entrado todo em minúsculas.

Exemplos

- Rodolfo Migon Favaretto => RMF
- Gléderson Lessa dos Santos => GLS
- Fulana de Tal => FT

- 7) Faça um programa que, a partir de um texto digitado pelo usuário, conte o número de caracteres total e o número de palavras (palavra é definida por qualquer sequência de caracteres delimitada por espaços em branco) e exiba o resultado.

- 8) Data por extenso. Faça um programa que solicite a data de nascimento (dd/mm/aaaa) do usuário e imprima a data com o nome do mês por extenso.

Exemplo

Data de Nascimento: 13/04/1988
Você nasceu no dia 13 de **abril** de 1988.

- 9) Dados uma frase e uma palavra, escreva um programa que verifique o número de vezes que a palavra ocorre na frase.

Exemplo

Na frase "ANA E MARIANA GOSTAM DE BANANA", a palavra ANA ocorre 4 vezes.

- 10) Leia uma cadeia de caracteres no formato "DD/MM/AAAA" e copie o dia, mês e ano para 3 variáveis inteiras. Antes disso, verifique se as barras estão no lugar certo, e se DD, MM e AAAA são numéricos.

11)O código de César é uma das mais simples e conhecidas técnicas de criptografia. É um tipo de substituição na qual cada letra do texto é substituída por outra, que se apresenta no alfabeto abaixo dela um número fixo de vezes. Por exemplo, com uma troca de três posições, ‘A’ seria substituído por ‘D’, ‘B’ se tornaria ‘E’, e assim por diante. Implemente um programa que faça uso desse Código de César (3 posições), entre com uma *string* e retorne a *string* codificada.

Exemplo

Entrada:

“A ligeira raposa marrom saltou sobre o cachorro cansado”

Saída:

“D OLJHLUD UDSRVD PDUURP VDOWRX VREUH R FDFKRUUR FDQVDGR”

12)Faça um programa que conte quantas vezes cada dígito aparece e qual é o conjunto de 5 dígitos consecutivos na sequência abaixo que gere o maior produto (multiplicação entre eles):

```
9349731671765313306249192251196744265747423553491949349698
3520312774506326239578318016984801869478851843858615607891
1294949545950173795833195285320880551112540698747158523863
0507156932909632952274430435576689664895044524452316173185
6403098711121722383113622298934233803081353362766142828064
4448664523874930358907296290491560440772390713810515859307
9608667017242712188399879790879227492190169972088809377665
7273330010533678812202354218097512545405947522435258490771
1670556013604839586446706324415722155397536978179778461740
6495514929086256932197846862248283972241375657056057490261
4079729686524145351004748216637048440319989000889524345065
8541227588666881164271714799244429282308634656748139191231
6282458617866458359124566529476545682848912883142607690042
```