

5ª Lista de Exercícios Cadeias de Caracteres

Para cada exercício, crie um programa em C/C++:

1. Leia uma string **s** e imprima o número de caracteres dessa string. Não use a função **strlen**.
2. Leia uma string **s** e imprima suas vogais e, em seguida, suas consoantes.
3. Leia uma string **s** e converta as letras maiúsculas em minúsculas. Dica: some 32 aos caracteres cujo código ASCII está entre 65 e 90 ('A'..'Z').
4. Leia uma string **s** e converta as letras minúsculas para maiúscula. Dica: subtraia 32 dos caracteres cujo código ASCII está entre 97 e 122 ('a'..'z').
5. Leia uma string **s** e um caracter **c** e imprima a posição de **c** em **s**; ou -1, caso **c** não pertença a **s**.
6. Leia uma string **s** e um caracter **c** e imprima o número de ocorrências de **c** em **s**.
7. Leia uma string **s** e dois caracteres **c1** e **c2**. Em seguida, substitua todas as ocorrências de **c1** por **c2** na string **s** e imprima a string resultante.
8. Leia uma string **s** e inteiros **i** e **j** e imprima o segmento de **s** de **i** até **j**. Não imprima nada se **i** ou **j** forem inválidos.
9. Leia uma string **s** e elimine seus espaços em branco "supérfluos", ou seja, reduza uma sequência de dois ou mais espaços em branco para apenas um espaço em branco. Exemplo:
"Essa cadeia possui vários espaços superfluos" → "Essa cadeia possui vários espaços superfluos"
10. Leia uma string **s** e um caracter **c**. Em seguida, remova todas as ocorrências do caracter **c** da string **s** e imprima **s**. Exemplo:
s: Programacao
c: a
Progrmco
11. Leia duas strings **s1** e **s2** e calcule a posição da string **s2** na string **s1**. Se não existir imprima -1. Ignore a diferença de maiúsculas e minúsculas. Exemplo:
s1 = "Sistemas de Informacao"
s2 = "info"
posição = 12

12. Leia duas strings **s1** e **s2** gere a string **s3** como resultado da intercalação de **s1** com **s2**. Imprima **s3**.
13. Leia duas strings **s1** e **s2** gere a string **s3** como resultado da concatenação de **s1** com **s2**. Imprima **s3**. Não use a função **strcat**.
14. Leia duas strings **s1** e **s2** e uma posição **p** ($p \geq 0$). Em seguida, insira a string **s2** na string **s1** na posição **p** e imprima **s1**. Caso a posição **p** seja inválida, apresente uma mensagem de erro.
15. Leia uma string **s1** e um número inteiro **n** ($n > 0$). Em seguida, gere a string **s2** com os **n** primeiros caracteres de **s1** e imprima **s2**.
16. Leia uma string **s1** e um número inteiro **n** ($n > 0$). Em seguida, gere a string **s2** com os **n** últimos caracteres de **s1** e imprima **s2**.
17. Leia uma string **s**, uma posição **p** ($p \geq 0$) e um valor **n** ($n > 0$). Em seguida, remova **n** caracteres da string **s** a partir da posição **p** e imprima **s**.
18. Leia uma string **s1** e gere a string **s2** com as mesmas letras de **s1** uma posição à frente, ou seja, a/A vira b/B, b/B vira c/C e z/Z vira a/A. Devem ser consideradas somente as letras de A a Z.

Exemplos:

Lua → Mvb

BOIS → CPJT

19. Leia uma string e imprimi-a como no exemplo abaixo:

Entrada:

AMOR

Saída:

AMOR

AMO

AM

A

20. Leia uma string e imprimi-a como no exemplo abaixo:

Entrada:

TERRA

Saída:

TERRA

TERR

TER

TE

T

TE

TER

TERR

TERRA

21. Leia uma string `s` informe se `s` é palíndroma ou não. Para a verificação devem ser consideradas somente as letras de A a Z. Um palíndromo é uma palavra ou frase, que é igual quando lida da esquerda para a direita ou da direita para a esquerda (espaços em brancos e pontuação devem ser descartados). Assuma que as palavras não tem acentos.

Exemplos:

Anotaram a data da maratona

A cara rajada da jararaca

A grama é amarga

Socorram-me, subi no ônibus em Marrocos!

A torre da derrota

Desafios

22. Leia uma string `msg` e imprima-a como um diamante, iniciando com o caracter central e, a cada linha, imprima os dois caracteres faltantes nas duas extremidades. Após imprimir a string completa, realize a operação inversa, até imprimir somente o caracter central. Exemplo:

Entrada:

FELIZ ANIVERSARIO

Saída:

```

      I
     NIV
    ANIVE
   ANIVER
  Z ANIVERS
 IZ ANIVERSA
LIZ ANIVERSAR
ELIZ ANIVERSARI
FELIZ ANIVERSARIO
ELIZ ANIVERSARI
LIZ ANIVERSAR
 IZ ANIVERSA
  Z ANIVERS
   ANIVER
    ANIVE
     NIV
      I
```

23. O algoritmo de César é um dos mais antigos e populares e é usado para criptografia de cadeias de caracteres (*strings*). Esse algoritmo usa um valor inteiro, chamado *chave*, e cada caracter da cadeia de caracteres é substituído por outro caracter à frente, de acordo com o valor da *chave*, levando-se em conta a ordem das letras no alfabeto. Por exemplo, se o valor da chave é 3 (três) e a cadeia de caracteres é “Programacao”, o resultado da criptografia é “Surjudpdfdr”, pois ‘S’ está 3 posições à frente de ‘P’ e ‘u’ está 3 posições à frente de ‘r’. Caso, o caracter à frente esteja além da letra ‘z’,

deve-se assumir que depois da letra 'z' o alfabeto reinicia com a letra 'a'. Assim, se o valor da chave é 8 (oito) e a cadeia de caracteres é "Tatu", o resultado da criptografia é "Bibc". Da mesma forma, os dígitos de '0' a '9' devem ser substituídos pelos dígitos à frente, sendo que depois do '9' reinicia em '0'. Note que deve ser respeitado o fato dos caracteres estarem em caixa baixa ou alta. Caracteres que não fazem parte do alfabeto regular (letras 'a' até 'z') ou não são dígitos ('0' a '9') não devem ser substituídos. Logo, se o valor da chave é 3 (três), a cadeia de caracteres "Programação C 2020" será criptografada como "Surjudpdçã F 5353". Leia uma string **msg** e uma chave (inteiro > 0) e imprima a string criptografada de acordo com o algoritmo de César.

24. Leia uma frase e imprima uma estatística mostrando os caracteres presentes na frase e as respectivas quantidades. Devem fazer parte da estatística somente as letras (A..Z) e dígitos (0..9). Letras miúsculas e minúsculas devem ser consideradas iguais. Exemplo:

Entrada:

O exercicio e facil

Saída:

A = 1

C = 3

E = 3

F = 1

I = 3

L = 1

O = 2

R = 1

X = 1