

Lista de Exercícios 9

Informações sobre cópias

As questões são individuais. Em caso de cópias de trabalho a pontuação será zero para os autores originais e copiadores. Não serão aceitas justificativas como: “Fizemos o trabalho juntos, por isso estão idênticos”.

Parte A – Exercícios para serem entregues

Crie funções/procedimentos recursivos para os exercícios a seguir., conforme solicitado em cada exercício.

Sugestão: construa um algoritmo em linguagem de programação C++ que tenha um menu principal para acessar as funções recursivas dos exercícios. Após realizar uma das operações indicadas no menu, o menu deverá voltar a ser exibido para que o usuário possa escolher uma nova opção. Lembre-se de cercar opções inválidas e criar uma opção para sair do menu.

- 1) Construa um procedimento recursivo que escreva na tela a sequência: 1, 2, 3, 4, 5,
- 2) Construa uma versão do procedimento anterior de forma que o limite superior seja parametrizado.
- 3) Construa um procedimento recursivo que receba uma string e printe a string reversa:
Ex.: recebe “ola mundo” e printa “odnum alo”
- 4) Construa um procedimento recursivo que receba uma string e diz se ela é ou não um palíndromo.
- 5) Crie um procedimento recursivo que printe os divisores de um número natural N.
- 6) Crie um procedimento recursivo que printe os divisores comuns entre dois números naturais N e M.
- 7) Crie uma versão recursiva para o algoritmo que trata o problema do cálculo da potenciação sem uso de função de potência (ex. pow).
- 8) Crie uma função recursiva que calcule o fatorial de um número natural N.
- 9) Construa algoritmos para calcular o k-ésimo termo Fibonacci utilizando as abordagens abaixo. Antes de iniciar a codificação, busque e planeje, cuidadosamente, as estratégias que o algoritmo deverá adotar.
 - (a) Algoritmo recursivo
 - (b) Algoritmo iterativo

Parte B – Exercícios para Treino – Não precisam ser entregues

1. Escreva um procedimento recursivo que recebe um vetor de comprimento N e o imprime na tela na ordem inversa de entrada.
2. Escreva um procedimento recursivo que recebe um vetor de comprimento N e o escreve em um arquivo (saida.txt) na ordem inversa de entrada.
3. Escreva uma função recursiva $divide(A,B)$ para calcular a divisão inteira de A por B utilizando apenas subtrações.
4. Escreva uma função recursiva que recebe um número inteiro e retorne o número de dígitos. Exemplo. $N = 50$ – saída deverá ser 2.
5. Escreva uma função recursiva que recebe um número inteiro e retorne a soma dos seus dígitos. Exemplo = $N = 45$. Saída deverá ser 9.
6. Escreva uma função recursiva que retorne o resultado da multiplicação dos dois parâmetros utilizando apenas o operador somas.
7. Escreva uma função recursiva que calcule o n -ésimo termo de uma progressão geométrica onde o primeiro termo e a razão são informados pelo usuário.
8. Escreva uma função recursiva que recebe um vetor como parâmetro e um valor k . Caso o valor k exista no vetor, a função deverá retornar a sua posição, senão deverá retornar -1.

Dado um vetor de inteiros A , apresente algoritmos recursivos para calcular:

9. O elemento máximo do vetor.
10. A soma dos elementos do vetor.
11. O produto dos elementos do vetor.
12. O máximo divisor comum (MDC) de dois números inteiros x e y pode ser calculado usando-se uma definição recursiva:

$$MDC(x, y) = MDC(x-y, y), \text{ se } x > y$$

Além disso, sabe-se que:

$$MDC(x, y) = MDC(y, x)$$

$$MDC(x, x) = x$$

Exemplo:

$$MDC(10, 6) = MDC(4, 6) = MDC(6, 4) = MDC(2, 4) = MDC(4, 2) = MDC(2, 2) = 2$$

Crie um algoritmo que leia os dois valores inteiros e utilize a função criada para calcular o MDC de x e y , e imprima o valor computado.

13. Gere recursivamente todas as possíveis combinações para uma senha com 4 dígitos inteiros.
14. Crie um algoritmo que leia uma palavra (vetor de caracteres) e gere recursivamente um arquivo para cada letra da palavra. Dentro de cada arquivo deverá ser gravada a letra e sua posição na palavra.