3ª Lista de Exercícios

Assunto - Estruturas de Repetição em Algoritmos e em Linguagem C

Essa lista de exercícios tem como objetivo principal desenvolver algoritmos a partir dos conteúdos abordados em sala de aula. Todos os exercícios também devem ser implementados em linguagem C.

1. **(Fácil)** Desenvolva um algoritmo que leia um número inteiro x e um inteiro não-negativo n, calcule e escreva o valor de x^n.

2. **(Fácil)** Desenvolva um algoritmo que leia um número inteiro positivo e escreva os divisores desse valor.

3. **(Fácil)** Desenvolva um algoritmo que escreva a seguinte árvore. O número de linhas deve ser fornecido pelo usuário.

   ```
   *
   **
   ***
   ****
   *****
   ******
   ```

4. **(Fácil)** Desenvolva um algoritmo que leia dois números inteiros. O primeiro número representa o número de capítulos e o segundo o número de seções. Usando esses valores o programa deve produzir uma saída como a de um a lista de capítulos e seções de um livro. Por exemplo, se as entradas forem 3 (capítulos) e 2 (seções), a saída deve ser:

   ```
   Capitulo 1
   .    Secao 1.1
   .    Secao 1.2

   Capitulo 2
   .    Secao 2.1
   .    Secao 2.2

   Capitulo 3
   .    Secao 3.1
   .    Secao 3.2
   ```

   Observe a indentação da saída.

5. **(Fácil)** Desenvolva um algoritmo que leia um número inteiro não-negativo n, calcule e escreva o valor de $n!$.

6. **(Fácil)** Desenvolva um algoritmo que leia a razão, o primeiro e último termo de uma PG e escreva:
   (a) Todos os elementos da PG nesse intervalo.
   (b) A soma dos elementos da PG nesse intervalo.

7. **(Fácil)** Cada um dos caracteres que são mostrados na saída do seu programa é representado internamente por um número, que varia entre 32 e 127. Desenvolva um algoritmo que mostre qual caractere corresponde a cada número.

8. **(Difícil)** Desenvolva um algoritmo que receba um numero inteiro n a ser elevado ao quadrado e escreva os dígitos do resultado por extenso. Exemplo: $n = 9$, resultado "Oito Um".

9. **(Fácil)** Desenvolva um algoritmo que gera números entre 1000 e 1999 e mostra aqueles que ao ser divididos por 11 dão resto 5.

10. **(Fácil)** Desenvolva um algoritmo que leia 10 valores reais, encontra o maior e o menor deles e escreva o resultado.

11. **(Médio)** O número 3025 possui a seguinte característica: $30 + 25 = 55$; $55^2 = 3025$. Desenvolva um algoritmo que escreva todos os números de 4 algarismos que apresentam tal característica.

12. **(Médio)** Desenvolva um algoritmo que calcule e escreva o n-ésimo termo da sequência de Fibonacci. O n-ésimo número da sequência de Fibonacci F_n é dado pela seguinte fórmula de recorrência:

$$F_n = \begin{cases} 1 & , se\ n = 1 \\ 1 & , se\ n = 2 \\ F_{n-1} + F_{n-2} & , se\ n > 2 \end{cases}$$

13. **(Médio)** Desenvolva um algoritmo que leia um número inteiro na base binária e o transforma para a base decimal.

14. **(Difícil)** Desenvolva um algoritmo que leia um número natural na base decimal, transformá-lo para a base binária.

15. **(Médio)** Desenvolva um algoritmo que simule uma calculadora. O programa receberá de entrada um inteiro que corresponde a quantos números serão lidos. Depois disso leia os N inteiros, porém entre um inteiro e outro há um símbolo que dirá qual operação deverá ser feita (as operações são todas entre números inteiros). Use a resposta da primeira conta para realizar a operação com o terceiro número e assim sucessivamente. As opções de operação são "+","−","/","∗" e "%". Ignore a precedência de operadores de multiplicação e divisão e faça as operações na ordem.
   Ex: 2 + 6 % 3 * 5
   = ((2+6) % 3) * 5 = (8 % 3) * 5
   Resultado: 10

16. **(Médio)** Desenvolva um algoritmo que calcula quantos metros um elevador percorre em um dia. Seu programa irá receber dois inteiros, o número de andares (N) em que o elevador parou e a distancia em metros entre dois andares sucessivos. Logo após leia os N andares em que o elevador parou e calcule a distância total que ele deslocou.
    Ex: 3 4 2 5 1
    Resposta:$(|2-5|*4)+(|5-1|*4)=12+16=28$

17. **(Médio)** Um dado comum possui 6 faces, numeradas de 1 a 6. Este dado, quando lançado inúmeras vezes,tende a sair o mesmo número de vezes para cada face. Isso não ocorre com um dado viciado, que tem uma tendência de resultar mais vezes em determinados números. Desenvolva um algoritmo que leia um inteiro N que representa o número de vezes que o dado foi lançado, e depois leia mais N inteiros que dizem qual foi o resultado de cada lançamento. Imprima a face que foi exibida com mais frequência e quantas vezes ela apareceu. Caso 2 ou mais faces tenham sido exibidas o mesmo número de vezes imprima o número de todas elas.
    Ex: 6 2 4 4 1 6 1
    Resultado: 1 4 2