

LISTA DE EXERCÍCIOS DE ALGORITMO – NÍVEL 2

Exercícios envolvendo estruturas de decisão

Fazer um algoritmo para:

- 1) Receber um número do usuário e mostrar se esse número é par ou não par
- 2) Receber 3 valores numéricos, X, Y e Z, e verificar se esses valores podem corresponder aos lados de um triângulo. Em caso afirmativo, informar ao usuário se o triângulo é equilátero, isósceles ou escaleno.
- 3) Receber 3 valores numéricos em 3 variáveis, A, B e C, e trocar os valores entre as variáveis de forma que, ao final do algoritmo, a variável A possua o menor valor e a variável C o maior.
- 4) Receber valores de base e altura de um triângulo e verificar se são valores válidos (positivos maiores que zero). Em caso afirmativo, calcular a área do triângulo.
- 5) Dado um número inteiro de segundos, mostrar a quantas horas, minutos e segundos ele corresponde.

Exercícios envolvendo estruturas de repetição

- 6) Escrever os N primeiros números da série de Fibonacci, onde N é um valor lido do usuário.

Série de Fibonacci: $n_1, n_2, n_3, \dots, n_m$ onde

$n_i = 1$, para $i \leq 2$

$n_i = n_{i-2} + n_{i-1}$, para $i > 2$

- 7) Calcular os números inteiros de quatro algarismos que possuem a mesma característica do número 3025.

$30+25 = 55$ e $55^2 = 3025$

Dica: utilizar a função $\text{FRAC}(n)$ que retorna a parte decimal de um número não inteiro

- 8) Ler um número do usuário e determinar se ele é ou não primo.

9) Calcular a soma da seguinte série de 100 termos:

$$1 - 1/2 + 1/4 - 1/6 + 1/8 - 1/10 + 1/12 \dots$$

10) Calcular o MDC de dois números inteiros usando o método de Euclides.

11) Calcular o volume de uma esfera em função do raio R. O raio deverá variar de 0 a 20 cm de 0,5 em 0,5 cm.

$$V = 4 \times \pi \times R^3 / 3$$

12) Calcular o número de grãos de milho que se pode colocar num tabuleiro de xadrez, colocando 1 no primeiro quadro e nos quadros seguintes o dobro do quadro anterior.

13) Faça um algoritmo para ler um número natural N e calcular o maior número primo menor do que o número N

14) Fazer um algoritmo que receba um número inteiro positivo N do usuário e mostre o valor do seu fatorial. Se o usuário não digitar um número inteiro positivo deve ser mostrada uma mensagem de erro e pedir para que ele digite novamente.

Exercícios envolvendo manipulação de literais

15) Inverter os caracteres contidos em um literal (observação: só é permitido usar as funções LEN, INI e FIM passadas em sala de aula)

16) Faça um algoritmo que:

- receba uma frase (literal) do usuário;
- receba um caracter qualquer do usuário (também na forma literal). Se o usuário digitar mais do que um caracter deve ser mostrada uma mensagem de erro e pedir para que ele digite novamente;
- calcule quantas vezes aquele caracter ocorre na frase digitada inicialmente e mostre ao usuário.

17) Faça um algoritmo que:

- receba uma frase (literal) do usuário;
- receba uma sequência qualquer de caracteres do usuário (também em forma literal);
- calcule quantas vezes aquela sequência de caracteres ocorre dentro da frase digitada.

Exemplo: se o usuário digitar a frase “A ARANHA ARRANHA A RÃ” e depois digitar a sequência “RA”, o algoritmo deve mostrar “O número de vezes que a sequência RA aparece na frase é 3” (considerar que o A acentuado é igual a um A normal)

Exercícios envolvendo modularização (funções)

18) Fazer a função POW (base, expoente) que recebe base e expoente como parâmetros e retorna o valor de base elevado a expoente. Se os valores de base e expoente não forem inteiros a função deve retornar 0.

19) Escreva uma função (somente a função!) MIX que receba dois valores literais como parâmetros e retorne o conteúdo dos dois literais intercalados, caracter a caracter.

Ex: MIX(“primeiro”, “segundo”) => “psreigmuenidroo”

20) Dado o seguinte algoritmo

```
declare Z numérico;  
função numérico OP ( X, Y )
```

```
declare X, Y numéricos;  
se Z = 0  
então
```

```
OP <- X+Y;
```

```
senão
```

```
OP <- X-Y;
```

```
fim se
```

```
fim função  
Algoritmo Exercício
```

```
declare X, Y numéricos;  
declare RES numérico;  
leia Z, Y, X;  
RES <- OP(Y, X);  
Escreva “O resultado de op sobre Y e X é”, RES;
```

```
fim algoritmo
```

Responda:

a) O que será mostrado para o usuário ao final do algoritmo se ele entrar com os seguintes valores em resposta ao comando de entrada:

1 -12 3

b) Por que o valor de Z é conhecido (e pode ser testado) dentro da função OP?

c) OP poderia ser reescrita como uma subrotina, mantendo-se o restante do algoritmo inalterado? Se não, explique por quê.

Exercícios envolvendo matrizes e vetores

21) Receber do usuário uma lista de N nomes e idades de pessoas, onde N também é fornecido pelo usuário, e mostrar o nome e a idade da pessoa mais idosa e da pessoa mais jovem.

22) Gerar a matriz transposta de uma matriz 5x5 dada pelo usuário (a transposta é obtida permutando-se as linhas e as colunas de uma matriz).

23) Queremos efetuar a compactação de um vetor V1 de N algarismos 0 e 1 ($N \leq 40$) digitado pelo usuário (onde cada algarismo ocupa uma posição do vetor), de forma que o vetor resultante V2 de N elementos ($N \leq 40$) possua menos elementos do que o vetor original. A regra de compactação é a seguinte:

- a) o primeiro elemento do vetor V2 é o número de algarismos zero que o vetor V1 contém, a partir do seu início, até o primeiro algarismo um;
- b) o próximo elemento do vetor V2 é o número de algarismos um que o vetor V1 contém, a partir do último zero encontrado, até o próximo algarismo zero;
- c) o próximo elemento do vetor V2 é o número de algarismos zero que o vetor V1 contém, a partir do último um encontrado, até o próximo algarismo um;
- d) repete-se os passos b) e c) até o final do vetor V1.

Exemplo: para o vetor digitado $V1 = (0,0,0,1,1,0,1,0,1,1,0)$
obtem-se $V2 = (3,2,1,1,1,2,1)$

Fazer o algoritmo para efetuar esta compactação, recebendo o valor de N e o vetor V1 do usuário e testando se os algarismos digitados são somente 0's e 1's.

24) Escreva um algoritmo que calcule a interseção (valores em comum) entre os valores contidos em dois vetores V1 e V2 e armazene estes valores no vetor V3.

25) Faça um algoritmo para receber do usuário 10 nomes de postos de gasolina e os preços da gasolina em cada um deles e depois mostrar os nomes e os

respectivos preços em ordem, do menor para o maior preço. (Dica: utilizar dois vetores para armazenar os nomes e os preços).

26) Faça um algoritmo para ordenar um vetor de N inteiros (Dica: encontrar o maior de todos os inteiros e trocá-lo com o último elemento do vetor, repetindo este procedimento N vezes porém descartando, a cada iteração, os elementos já ordenados).

27) Uma matriz quadrada A, cujos elementos são designados por $A_{i,j}$, é dita simétrica se $A_{i,j} = A_{j,i}$. Construir um algoritmo para ler uma matriz NxN do usuário (onde o valor de N também é fornecido pelo usuário) e determinar se ela é ou não simétrica.

Obs: funções prontas que podem ser utilizadas:

Funções de manipulação de literais:

LEN(x) – retorna o número de caracteres do literal x

INI(x, n) – retorna um literal que consiste nos n primeiros caracteres do literal x

FIM(x, n) – retorna um literal que consiste nos n últimos caracteres do literal x

Operador de concatenação |

Funções de manipulação de inteiros

TRUNCA(x)