

Exercícios trabalhados na aula

Modularização

Exercício 25:

Construa uma função capaz de receber um número inteiro como parâmetro e retornar se este é ou não um número ímpar.

Resposta:

```
funcao eh_imp(ar: inteiro): logico
inicio
    retorne (a%2<>0)
fimfuncao
```

Modularização

Exercício 25:

Resposta alternativa:

```
funcao eh_impar (a: inteiro): caractere
inicio
    se (a%2<>0) entao
        retorne ("O número é impar.")
    senao
        retorne ("O número é par.")
    fimse
fimfuncao
```

Modularização

No caso em que o problema anterior se apresentasse da seguinte forma: Construa uma função que receba um número inteiro e retorne se este é ou não um número ímpar. Algo mudaria com relação à interpretação?

Sim. Neste caso não estaria especificado se a função receberia o número através de um parâmetro ou se este seria lido através da entrada padrão. Logo, a seguinte solução também seria correta:

```
funcao eh_impair (): logico
var n:inteiro
inicio
```

```
    escreva ("Entre com um valor inteiro: ")
```

```
    leia (n)
```

```
    retorne (n%2<>0)
```

```
fimfuncao //a chamada da função eh_impair deve ser
           //feita da seguinte forma: eh_impair()
```

Modularização

Exercício 26

Utilizando-se do conceito de modularização construa um algoritmo que resolva o problema de obter as raízes reais de uma equação do segundo grau, caso existam raízes reais.

Modularização

Exercício 27

Construa uma função que receba, como parâmetro, um número inteiro positivo, o qual representa a posição de um determinado termo na série de Fibonacci, a função deve retornar o valor do termo correspondente à posição recebida.

Modularização

Exercício 28

Construa um algoritmo que tenha a capacidade de efetuar uma multiplicação entre valores naturais quaisquer e a capacidade de calcular o fatorial de um número natural qualquer. Tanto no cálculo da multiplicação quanto no cálculo do fatorial os únicos operadores aritméticos que podem ser utilizados são os de soma e subtração.

O algoritmo em questão deve possibilitar ao usuário fazer a seleção de qual operação será realizada. As entradas devem ser validadas e o conceito de modularização deve ser aplicado.