

Exercícios de Fixação - Lógica de Programação

Bóson Treinamentos em Tecnologia

Sub-rotinas: Funções e Procedimentos

Crie os algoritmos para os itens a seguir, testando os programas no Portugol Studio.

Exercício 1

Crie um programa que use uma função para calcular e retornar a área em m² de um terreno retangular, recebendo como argumentos os valores de largura e comprimento desse terreno.

A área do retângulo é dada por:

$$A = l \times c$$

Onde *l* é a largura e *c* o comprimento, em metros.

Exercício 2

A série de fibonacci é uma sequência de números que inicia com 0 e 1 e na qual cada número subsequente é a soma dos dois números anteriores a ele, da seguinte maneira, infinitamente:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Trata-se de uma série numérica que ocorre na natureza e descreve uma forma espiral. A razão entre dois fibonaccis sucessivos converge para o valor 1,618..., chamado de "Proporção Áurea", esteticamente muito agradável.

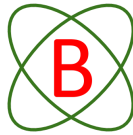
Crie um algoritmo que empregue uma função para obter o *n-ésimo* número de fibonacci de uma sequência, fornecido pelo usuário.

Por exemplo, o 3º fibonacci é o número 1 e o 8º fibonacci é o valor 13.

Exercício 3

Crie uma função que permite converter um número decimal, fornecido pelo usuário, e seu equivalente binário.

Por exemplo, o número 65 (decimal) equivale a 1000001 em binário.



Dica: Use a técnica de divisões sucessivas por 2 para realizar a conversão, e empregue o operador de módulo para coletar os restos de cada divisão, como ilustra a imagem de exemplo a seguir:

Conversão de Decimal para Binário

49	÷	2	=	24	Resto (sobra)	
24	÷	2	=	12	Resto (sobra)	
12	÷	2	=	6	Resto (sobra)	
6	÷	2	=	3	Resto (sobra)	
3	÷	2	=	1	Resto (sobra)	
1	÷	2	=	0	Resto (sobra)	

As divisões por 2 param quando
o quociente chega a 0

$$49_{10} = 110001_2$$

Saída de exemplo:

```
Programa que converte um número decimal para binário.  
Digite um número decimal: 49  
O número 49 em binário é 110001
```

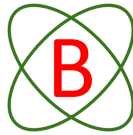
Exercício 4

Um número primo é um número inteiro positivo que possui apenas dois fatores: um e ele próprio.

Ou seja, é um valor divisível apenas por si mesmo e por um.

Escreva um programa que receba um número digitado pelo usuário e diga se o número é primo ou não, usando uma função.

Dica: Use o operador de módulo para verificar se o número fornecido é primo.



Lista de números primos entre 1 e 100:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 e 97.

Exemplo de saída esperada:

```
Números Primos
Entre com um número para verificar se ele é primo:
89

O número 89 é primo.
```

Exercício 5

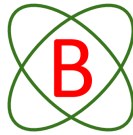
Crie uma função que receba um vetor de 10 números inteiros como argumento e retorne o maior valor armazenado nesse vetor.

Para acelerar os testes, preencha o vetor com números aleatórios entre 1 e 100.

Saída de exemplo:

```
Verificar maior valor armazenado em um vetor

O maior valor do vetor é 97
Números armazenados: 66 76 37 8 18 10 49 97 60 38
```



Resolução dos Exercícios

Exercício 1

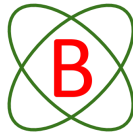
```
programa
{

    real largura = 0.0
    real comprimento = 0.0
    real areaTotal = 0.0

    funcao inicio()
    {
        escreva("Informe a largura do terreno: ")
        leia(largura)
        escreva("Qual o comprimento do terreno?: ")
        leia(comprimento)

        areaTotal = calculaArea(largura, comprimento)
        escreva("A área do terreno é de " + areaTotal + " m²")
    }

    funcao real calculaArea(real l, real c) {
        real area = 0.0
        area = l * c
        retorne area
    }
}
```

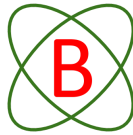


Exercício 2

```
programa
{
    inteiro num

    funcao inicio()
    {
        escreva("Obter enésimo Fibonacci\n")
        escreva("Entre com um número: ")
        leia(num)
        escreva(num + "º Fibonacci: " + fibonacci(num))
    }

    // Função fibonacci
    funcao inteiro fibonacci(inteiro x) {
        se (x <= 1) {
            retorne x
        }
        retorne (fibonacci(x-1) + fibonacci(x-2))
    }
}
```



Exercício 3

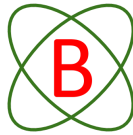
```
programa
{

    inteiro dec, bin
    funcao inicio()
    {
        escreva("Programa que converte um número decimal para binário.\n")
        escreva("\nDigite um número decimal:\n")
        leia(dec)

        // Invocar a função
        bin = binario(dec)

        escreva("\nO número " + dec + " em binário é " + bin)
    }

    // Função que converte de decimal para binário
    funcao inteiro binario(inteiro numDec) {
        inteiro numBin=0, resto, f=1
        enquanto(numDec != 0) {
            resto = numDec % 2
            numBin = numBin + resto * f
            f = f * 10
            numDec = numDec / 2
        }
        retorne numBin
    }
}
```

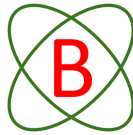


Exercício 4

```
programa
{
    inteiro num, res
    funcao inicio()
    {
        escreva("Números Primos\n")
        escreva("Entre com um número para verificar se ele é primo:\n")
        leia(num)

        res = primo(num)
        se(res == 1) {
            escreva("\nO número " + num + " é primo.\n")
        }
        senao {
            escreva("\nO número " + num + " não é primo.\n")
        }
    }

    funcao inteiro primo(inteiro n) {
        inteiro i = 2
        se(n < 2) {
            retorne 0
        }
        enquanto (i <= n/2) {
            se (n % i == 0) {
                retorne 0
            }
            senao {
                i++
            }
        }
        retorne 1
    }
}
```



Exercício 5

```
programa
{
    inclua biblioteca Util -->u
    inteiro maior, cont
    inteiro numeros[10]

    funcao inicio()
    {
        escreva("Verificar maior valor armazenado em um vetor\n")
        // Preencher o vetor com valores aleatórios
        para(cont = 0; cont < 10; cont++) {
            numeros[cont] = u.sorteia(1,100)
        }

        maior = maiorValor(numeros)

        escreva("\nO maior valor do vetor é " + maior)

        // Opcional: mostrar o vetor preenchido
        escreva("\nNúmeros armazenados: ")
        para(cont = 0; cont < 10; cont++) {
            escreva(numeros[cont] + " ")
        }
        escreva("\n")
    }

    funcao inteiro maiorValor(inteiro vetorN[]) {
        inteiro i = 1, max
        max = vetorN[0]
        enquanto(i < 10)
        {
            se (max < vetorN[i]) {
                max = vetorN[i]
            }
            i++
        }
        retorne max
    }
}
```