

2ª Lista de Exercícios – 29/10/2015 (entregar 05/11/2015)

Exercício 1: Faça um programa em C para exibir os Números Primos do intervalo informado. Utilizar as funções `eh_primo()` e `eh_divisor()`.

Exercício 2: Faça um programa em C para verificar se um número é Primo ou De Armstrong. Utilizar as funções `eh_primo()` e `eh_armstrong()`.

Exercício 3: Faça um programa em C para verificar se um número pode ser expresso como a soma de dois números primos. Utilizar as funções `eh_primo()` e `eh_soma_primos()`.

Exercício 4: Faça um programa em C para encontrar a soma dos fatoriais dos números naturais de um intervalo informado. O intervalo pode ser crescente ou decrescente. Utilizar a função `fatorial()`.

Exercício 4: Faça um programa em C para encontrar o MDC de vários pares de inteiros positivos. O programa pára quando encontra um par de números iguais. Utilizar as funções `mdc()` e `eh_divisor()`.

Exercício 5: Faça um programa em C para encontrar o MMC de N pares de inteiros positivos. Utilizar as funções `mmc()` e `eh_divisor()`.

Exercício 6: Faça um programa em C para converter um número Binário para Decimal e vice-versa. O programa receberá o número e sua base (2 ou 10) . Utilizar as funções `bin_para_dec()` e `dec_para_bin()`.

Exercício 7: Faça um programa em C para converter um número Octal para Decimal e vice-versa. O programa receberá o número e sua base (8 ou 10) . Utilizar as funções `oct_para_dec()` e `dec_para_oct()`.

Exercício 8: Faça um programa em C para converter vários números Binários para Octais e vice-versa. Cada número lido virá acompanhado de sua base (2 ou 8). O programa pára quando encontra um número informado que contenha um dígito que não pertence ao sistema de numeração da base. Utilizar as funções de conversão já feitas e mais a função `percebe_a_base()`.

Exercício 9: Faça um programa em C para inverter os dígitos de vários números Binários, Octais e/ou Decimais. Cada número será precedido por sua base. Utilizar funções já definidas e mais as funções `inverte_digitos()` e `eh_palindromo()`. O programa pára após mostrar um número invertido palíndromo.

Exercício 10: Faça um programa em C para calcular a potência de vários números usando uma função `pot()`. A base do número pode ser inteira ou real e o expoente sempre será inteiro. O programa pára quando a valor da base for igual ao expoente.

Exercício 11: Faça um programa em C para verificar se um número é par ou ímpar. Utilizar uma função `par_ou_impár()` que retorna 1 para um número ímpar recebido e 2 para um par.

Exercício 12: Faça um programa em C para verificar se uma letra recebida é uma vogal ou consoante. O programa pode receber uma letra ou não. Caso não seja uma letra, será emitida uma mensagem adequada. Utilizar uma função `consoante_ou_vogal()` que retorna: 1 para vogal, 2 para consoante e 0 para outro caractere.

Exercício 13: Faça um programa em C para encontrar o maior número entre três números inteiros. Utilizar uma função `maior_de_treis()` que retorna o maior de três números inteiros recebidos.

Exercício 14: Faça um programa em C para encontrar as raízes de uma equação do segundo grau. Utilizar uma função `delta()`.

Exercício 15: Faça um programa em C para mostrar os anos bissextos de um intervalo de anos. Os anos que limitam o intervalo poderão estar em ordem crescente ou decrescente. Utilizar uma função `eh_bissexto()` que retorna verdadeiro ou falso.

Exercício 16: Faça um programa em C para verificar se um número real é positivo ou negativo ou zero. Utilizar uma função `sinal()` que retorna -1, 0 ou 1 para valores negativos, zero ou positivos, respectivamente, para o valor recebido.

Exercício 17: Faça um programa em C para verificar se um caracter recebido é alfabético ou não. Utilizar uma função `eh_letra()`.

Exercício 18: Faça um programa em C para Programa de C para calcular a soma dos Números Naturais até N. Utilizar uma função `soma_nat()` que recebe o valor de N e retorna a soma dos naturais até N.

Exercício 19: Faça um programa em C para encontrar fatorial de um número inteiro não negativo. Utilizar as funções `fatorial()` para retornar o fatorial de um numero e `nao_eh_negativo()` que retorna verdadeiro ou falso.

Exercício 20: Faça um programa em C para gerar uma tabelas de multiplicação do intervalo de inteiros entre 0 e N, crescente. N será lido. Cada tabela apresenta um número do intervalo multiplicado por números inteiros entre 1 e 10. Use a função `mostra_tabuada()` que recebe um inteiro e mostra a tabuada deste número multiplicado até 10.

Exercício 21: Faça um programa em C para exibir a Série de Fibonacci com números menores que N. Utilizar uma função `calcula_proximo_fibo()` que calcula o próximo número de Fibonacci da série e outra `eh_valido_n()` , para verificar se N é ou não maior que 1.

Exercício 22: Faça um programa em C para exibição de caracteres de A a Z, maiúsculos e minúsculos, utilizando único laço que gera valores de 0 a 255 (representação numérica dos caracteres). Utilizar a função `eh_letra()` já definida no exercício 17.

Exercício 23: Faça um programa em C para contar número de dígitos significativos de um inteiro positivo. Utilizar a função `conta_digitos()` e a função `eh_positivo()`, para dar mensagem quando o número é inválido.

Exercício 24: Faça um programa em C para exibir o número Armstrong entre dois números limites de em intervalo de inteiros. Os limites poderão ser fornecidos em qualquer ordem, crescente ou decrescente. Utilizar a função `eh_armstrong()` do exercício 2.

Exercício 25: Faça um programa em C para exibir fatores primos de um número (ver <http://www.somatematica.com.br/fundam/decomp.php>). Utilizar as funções `eh_divisor()` e `eh_primo()` já definidas para exercício 1.

Exercício 26: Faça um programa em C para fazer uma calculadora simples para adicionar, subtrair, multiplicar ou dividir, usando `switch ... case`. Utilizar uma função `operar()` que recebe dois operandos reais e o símbolo da operação (+, -, * ou /) e retorna o valor resultante da operação.

continua...