**Universidade Federal de Campina Grande**

**Centro de Engenharia Elétrica e Informática – CEEI**

**Departamento de Sistemas e Computação – DSC**

**Disciplina: Introdução à Programação – TURMA-02 - Manhã**

**2ª Lista de Exercícios ENTREGA: 26/Junho/15**

**Exercício 1:**Faça um programa que leia vários números na ordem crescente até que seja lido  um número menor que o anterior. O programa deve informar quantos números ordenados foram lidos.

**Exercício 2:** Faça um programa que leia vários inteiros quaisquer (não importa a quantidade de dígitos) e para cada número lido informe se é ou não um Número de Armstrong.  O programa encerra quando for lido um número não positivo.

**Exercício 3:** Faça um programa que leia vários números inteiros e para cada número lido informe seus divisores.  O programa encerra quando for lido um número não positivo e deve usar a função *imprime\_divisores()* que recebe como parâmetro um número inteiro e imprime os divisores do número recebido. PRAZO: 09/06/15.

**Exercício 4:** Faça um programa que imprima os Números de Armstrong existentes no intervalo [1..N].  A função *main()*do seu programa deve ser a que segue e, além além desta o programa deve usar as funções especificadas abaixo:

int main (){

int numero, n;

printf("Programa que imprime os Numeros Armstrongs existentes no intervalo [1..N]");

printf("Digite o valor de N: ");

scanf("%d", &n);

for (numero = 1; numero <= n; numero++)

if (eh\_num\_armstrong(numero))

printf("\n%d", numero);

return 0;

}

***conta\_digitos\_numero()*:** que recebe como parâmetro um número inteiro e retorna para o ponto de chamada a quantidade de dígitos desse número. (ex. 153 tem 3 dígitos)

***soma\_digitos\_potencia()*:** que recebe como parâmetros um número inteiro e a quantidade de dígitos desse número e retorna para o ponto de chamada o soma de cada dígito do numero elevado a quantidade. (ex. 153 = 1^3  + 5^3 + 3^3)

***eh\_num\_armstrong ()*:** que recebe ´como parâmetro um número inteiro e retorna para o ponto de chamada verdadeiro (1) se o número é um Num. de Armstrong, caso contrário retorna falso (0).

**Exercício-5:**  Na inauguração de uma loja, o empresário irá sortear 6 brindes entre os 10 primeiros clientes que chegarem na loja. Portanto distribuirá para os dez primeiros clientes fichas numeradas de 1 a 10.  Faça um programa para simular o sorteio das seis fichas que serão premiadas. A mesma ficha não poderá ser sorteada mais de uma vez.

Use a função ***imprime\_senhas\_sorteadas()*** que deve imprimir as seis fichas premiadas. Esta função deve usar a função ***sorteia\_ficha()***que retorna um valor aleatório no intervalo [1..10]. Observe que se ela retornar um valor que já foi sorteado, terá de ser ativada novamente.

**Exercício-6:** Refaça o código do programa [04\_11\_menu\_figuras\_triangulos.cpp](http://www.dsc.ufcg.edu.br/~intprog/Periodo-2015.1/exemplos_2015.1/04_11_menu_figuras_triangulos.cpp) apresentado no  *link* Exemplos das Aulas da página da disciplina. O novo código deverá usar uma função para cada tarefa. Tais como:  mostra\_menu(), mostraFigPadrao\_1( ), mostraFigPadrao\_2() , etc.

**Exercício-7:** Faça um programa que leia vários números inteiros até que seja lido um número negativo. Para cada número lido, o programa deve imprimir quantos dígitos zeros o número tem. Exemplos: 187263 tem 0 dígitos zero; 100 tem 2 dígitos zero; 0 tem um dígito zero; 500000604 tem 6 dígitos zero.

**Exercício-8:** Faça um programa que leia vários números inteiros positivos e informe, para cada número, uma mensagem dizendo se ele é ou não palíndromo. O programa parará quando encontrar um número inteiro negativo. Palíndromo é um número inteiro que se invertermos seus dígitos resulta no mesmo número (Ex.: 12321, 13577531, 11, 909, 8448, 7, etc.). O programa deve usar a função *eh\_palindromo(),* que recebe um inteiro positivo e retorna verdadeiro ou falso. E essa função deve usar a função *inverte\_num()* que recebe como parâmetro um inteiro e retorna esse inteiro com os dígitos invertidos (Ex.: recebe 1234 e retorna 4321; Recebe 100 e retorna 1)

Dica: Veja o programa [04\_06\_inteiro\_invertido.cpp](http://www.dsc.ufcg.edu.br/~intprog/Periodo-2015.1/exemplos_2015.1/04_06_inteiro_invertido.cpp) apresentado no  *link* Exemplos das Aulas da página da disciplina.

**Exercício-9:** Faça um programa para imprimir uma Tabela de conversão de Número Decimal para Número Binário no intervalo de 0 a 20. O programa deve usar a função *converteDec\_Binario()* que recebe o número em decimal e retorna o número binário correspondente.

**Exercício-10:** Faça um Programa que leia N números inteiros representando números binários e os converta para inteiros decimal. O programa deve utilizar a função *converteBin\_Decimal()* que recebe como parâmetro um inteiro representando um número binário e retorna o inteiro decimal correspondente ao binário recebido.

**Lembrar forma de entrega! Exemplo:**

**Para:** [**intprog.exercicios.t01@gmail.com**](mailto:intprog.exercicios.t02@gmail.com)

**Assunto:** L02-Exercícios 1 e 2 – nome do aluno (ou dupla)

**Anexo:**Um único arquivo *zipado* - o nome desse arquivo deve ser o nome do aluno (ou alunos, no caso de dupla)  – este arquivo *zipado* deverá conter os programas fonte (arquivos.cpp) e o nome de cada programa deve ser o número do exercício correspondente.

**Texto:**  Matrícula e nome do aluno (ou alunos, no caso de dupla)