



Faculdade de Computação

Programação Procedimental

1ª Lista de Exercícios p/ Avaliação

Prof. Cláudio C. Rodrigues

Instruções:

1. Apresentar as soluções usando a *linguagem C*, quando for apropriado;
2. A atividade deve ser desenvolvida em grupo de até 5 (um até cinco) estudantes e qualquer identificação de plágio sofrerá penalização;
3. As respostas para as atividades devem ser entregues na forma de arquivos gravados em CD/DVD, impreterivelmente, no dia **02/06/2017**. O CD/DVD deve conter arquivo com a identificação dos integrantes do grupo: nomes, números de matrícula e e-mail.

Problemas:

- P1)** O salário inicial de um engenheiro de software (ES) recém-formado é de R\$ 90.000,00 ao ano. Considere um índice de inflação anual de 8%, bônus de 4% a cada quatro anos trabalhados e um teto salarial para engenheiros de software de R\$ 150.000,00. Escreva um programa em linguagem C que leia o nome de um ES e o número de anos trabalhados e informe o salário atual. A saída deve parecer como:

João da Silva trabalha por xx anos. Salário atual: R\$ yyy,yy / ano

O algoritmo deve estar contido no arquivo "salario.c".

- P2)** No século I D.C., os números inteiros positivos dividiam-se em três categorias:
- **Perfeitos** – aqueles que são iguais à soma dos seus divisores. (p.ex. $6 = 1 + 2 + 3$)
 - **Abundantes** – inferiores à soma dos seus divisores. (p.ex. $12 < 1 + 2 + 3 + 4 + 6 = 16$)
 - **Reduzidos** – superiores à soma dos seus divisores. (p.ex. $9 > 1 + 3 = 4$)
- Note-se que, neste caso, eram considerados divisores de um número a unidade, mas não o próprio número.
- a) Elabore um programa em linguagem C que imprima a lista dos inteiros de 1 até 200, classificando-os em **perfeitos**, **abundantes** ou **reduzidos**. Calcule também a quantidade de elementos de cada categoria.

O algoritmo deve estar contido no arquivo "perfeitos.c".

- P3)** Calculando raiz quadrada: Os computadores (e qualquer espécie de equipamento eletrônico) somente conseguem realizar as quatro operações aritméticas básicas. Combinando essas operações é possível calcular funções mais complexas, como seno, cosseno e a raiz quadrada. Para calcular a raiz quadrada, os computadores calculam iterativamente, até conseguir um valor próximo o suficiente da raiz quadrada, mas nunca o valor exato. O cálculo é interrompido quando o erro é pequeno o suficiente para não ser percebido pelo usuário (ou seja, o erro não pode ser mais representado por uma variável de ponto flutuante). Ao atingir esse estado, apesar de ser possível calcular um valor mais preciso para a raiz quadrada, os computadores não podem mais armazená-lo.

Para o cálculo da raiz quadrada de um determinado valor **n**, a equação iterativa é:

$$x_{i+1} = \frac{1}{2} \left(x_i + \frac{n}{x_i} \right)$$

onde x_i é o valor atual da raiz quadrada e x_{i+1} é o próximo valor da raiz quadrada.

A primeira raiz (x_0) pode ser um valor qualquer, preferencialmente próximo da raiz quadrada (metade do valor é uma boa aproximação). Escreva uma função em C que, dado um número ponto flutuante, retorne a sua raiz quadrada.

O algoritmo deve estar contido no arquivo "raizquad.c".

- P4)** Verificar se um dado número natural é ou não um Número Primo.
 Teorema: Se um dado número natural **n** não possuir divisores $\leq \text{raizq}(n)$, então **n** é um número Primo.
raizq(n) → significa raiz quadrada de **n**
 O algoritmo deve estar contido no arquivo "ehprimo.c".

- P5)** Camponeses russos tem uma maneira interessante de efetuar a multiplicação de dois valores inteiros (não muito grandes). O método pode ser ilustrado da seguinte maneira. Considere o produto dos seguintes valores, 17 e 19. Esses valores serão colocados no topo de duas colunas (coluna esquerda e coluna direita). O número da coluna esquerda é sucessivamente dividido por 2 (divisão inteira), enquanto o número da coluna direita é sucessivamente multiplicado por 2. Os resultados são escritos um abaixo do outro em suas respectivas colunas. O processo é repetido até que o número da coluna esquerda contendo a divisão alcance o valor 1. Para calcular o resultado some os valores da coluna direita, se o número correspondente da coluna esquerda for ímpar. Veja o exemplo abaixo (17x19):

Divide por 2	Multiplica por 2
17	19
8	38
4	76
2	152
1	304

Assim, somando os números da coluna direita que tenham números ímpares na coluna esquerda, temos: **19 + 304 = 323**, que é igual ao produto de **17** por **19**.

Escreva um programa, em linguagem C, denominado **multi_russa** que receba como parâmetro de entrada dois números inteiros e positivos (**a** e **b**) e efetua o produto **axb** através do método descrito (*Russian Multiplication*). O programa deve apresentar no dispositivo padrão de saída (tela de display) o resultado da multiplicação obtido.

- P6)** A *conjectura de Collatz* é uma conjectura matemática que recebeu este nome em referência ao matemático alemão Lothar Collatz, que foi o primeiro a propô-lo, em 1937. Além desse nome, este problema também é conhecido como **Problema $3x + 1$** .

Esta conjectura aplica-se a qualquer número natural, e diz-nos para, se este número for par, dividi-lo por 2, e se for ímpar, multiplicá-lo por 3 e adicionar 1 ($3x+1$). Repetir esse processo até que o número se torne 1. Desta forma, por exemplo, se a sequência iniciar com o número 5 ter-se-á: 5; 16; 8; 4; 2; 1. A conjectura apresenta uma regra dizendo que, qualquer número natural, quando aplicado a esta regra, eventualmente sempre chegará a 4, que se converte em 2 e termina em 1.

Escreva um programa, em linguagem C, que leia do dispositivo padrão de entrada (*stdin*) um número natural **n** e escreva no dispositivo padrão de saída (*stdout*) a sequência de Collatz. Considere a seguinte operação em um número arbitrário **n** (inteiro positivo):

- Se o número **n** for par, divida-o por 2;
- Se o número **n** for ímpar, multiplique-o por 3 e some 1 ($3x+1$).

Exemplo:

n ? 6

sequencia: 6 3 10 5 16 8 4 2 1

- P7)** Escreva um programa, em linguagem C, denominado **eh_semente** que receba como parâmetros dois números inteiros e positivos X e Y. A função deve apresentar como resposta se o número X é dito ser um número semente de Y, ou não. Um número X é dito ser um "número semente" de um número Y se multiplicarmos X pelos seus dígitos e o resultado for igual a Y.
 Por exemplo, **123** é dito ser a "semente do número 738, pois, $123*1*2*3 = 738$.

- P8)** Escreva um programa, em Linguagem C, denominado **raiz_dezena** que imprima todos os milhares (4 algarismos – [1000,9999]) cuja raiz quadrada seja a soma das dezenas formadas pela divisão acima. Qualquer número natural de quatro algarismos pode ser dividido em duas dezenas formadas pelos seus dois primeiros e dois últimos dígitos. Por exemplo, 1297 pode ser separado nas 12 e 97.

Exemplo: raiz de 9801 = 99 = 98 + 01.

Portanto 9801 é um dos números a ser impresso

- P9)** Escreva um programa, em linguagem C, que construa uma pirâmide de números conforme ilustrado na figura abaixo. O programa deve ler do usuário a informação da altura da pirâmide **h**. Por exemplo, para uma altura **h=6** a pirâmide deve ser construída como na figura abaixo (automatize o algoritmo):

```

      1
    1 2 1
  1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 5 6 5 4 3 2 1

```

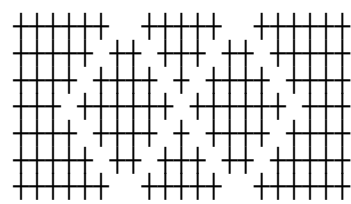
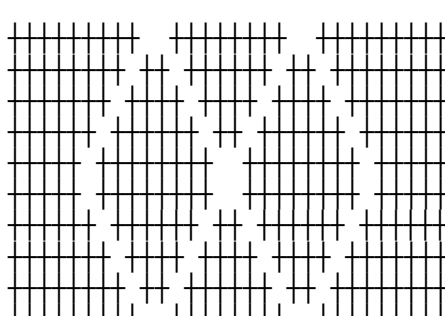
O algoritmo deve estar contido no arquivo "piramide.c".

- P10)** Escreva um programa, em linguagem C, que escreva no dispositivo padrão de saída (tela) a figura de um losango, como mostrado nas figuras abaixo. A dimensão da altura (**h**) deve ser informada pelo usuário (número ímpar) e o algoritmo deve se adaptar as escolhas do usuário.

h = 3	h = 5	h = 7
<pre> * * * * </pre>	<pre> * * * * * * * * </pre>	<pre> * * * * * * * * * * * * * * </pre>

O algoritmo deve estar contido no arquivo "losango.c".

- P11)** Escreva um programa, em linguagem C, que escreva no dispositivo padrão de saída (tela) a figura com o padrão apresentado nos exemplos abaixo. O padrão possui as dimensões **n** por **3n** – altura **n** com largura **3n** - A dimensão da altura **n** deve ser informada pelo usuário e o algoritmo deve se adaptar as escolhas do usuário e desenhar o padrão com o símbolo (+) definido na tabela ascii estendida de código **0xC5**.

n ? 7	n ? 10
	

O algoritmo deve estar contido no arquivo "infinito.c".

- P12)** Escreva em linguagem de programação C um programa denominado TapeteAlpha que receba como entrada o número (inteiro) de caracteres a ser impresso no formato de moldura. A moldura de caracteres deve ter a forma do exemplo abaixo. O padrão deve ser impresso diretamente na tela, sem utilizar armazenador em memória (por exemplo, memória).

Exemplo:

>Numero de caracteres: 3 >Saída: AAAAAA ABBBBA ABCCBA ABCCBA ABBBBA AAAAAA	>Numero de caracteres: 5 >Saída: AAAAAAAAAA ABBBBBBBBA ABCCCCCBA ABCDDDDCBA ABCDEEDCBA ABCDEEDCBA ABCDDDDCBA ABCCCCCBA ABBBBBBBBA AAAAAAAAAA
---	---

P13) Calcular e^x por desenvolvimento em Série de Taylor desprezando termos, em grandeza, inferiores a 10^{-7} .

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, \quad -\infty < x < \infty$$

A função exponencial deve somar os termos da série até aparecer um termo cujo valor absoluto seja menor que 0.0001 (precisão). Isto é, o termo $\left|\frac{x^k}{k!}\right|$ tende a zero quando k tende a $+\infty$. Repetir o cálculo dos termos até um k tal que $\left|\frac{x^k}{k!}\right| < 0.0000001$.

Dica: evite calcular o valor do fatorial, calcule um termo da série usando o termo anterior.

O algoritmo deve estar contido no arquivo "exponencial.c".

Análise Infinitesimal:

Desenvolvimento em série de Taylor da função exponencial:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots, \text{convergente } \forall x \in \mathbb{R}$$

Somas parciais de ordem n e $n + 1$:

$$S_n = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

$$S_{n+1} = S_n + \frac{x^{n+1}}{(n+1)!}$$

Termos de ordem n e $n + 1$:

$$t_n = \frac{x^n}{n!}$$

$$t_{n+1} = \frac{x^{n+1}}{(n+1)!} = t_n \frac{x}{n+1}$$

A série é convergente $\forall x \in \mathbb{R}$, isto é:

$$\lim_{n \rightarrow \infty} \{S_n\} = e^x$$

E, nesse caso:

$$\lim_{n \rightarrow \infty} \{t_n\} = 0$$

- P14)** Escreva um programa, em linguagem C, denominado *senox.c*, que calcula o seno de x . A sua função apenas deverá calcular os primeiros 10 termos, isto é, até ao termo $x^{19}/19!$. O seno de x pode ser calculado utilizando a expansão da série de Taylor:

$$\text{seno } x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

O elemento x da série de Taylor deve ser em radianos. Portanto, o programa *senox.c* deve ler o valor do ângulo em graus e converter para radianos. Para realizar a conversão utilize a regra de três, considerando que PI radianos é igual a 180° e que a constante PI assume valor de 3.141592

- P15)** Escreva um programa, em linguagem C, denominado *cosex.c*, que calcula o cosseno de x . A sua função apenas deverá calcular os primeiros 10 termos, isto é, até ao termo $x^{20}/20!$. O cosseno de x pode ser calculado utilizando a expansão da série de Taylor:

$$\text{cosseno } x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$$

O elemento x da série de Taylor deve ser em radianos. Portanto, o programa *cosex.c* deve ler o valor do ângulo em graus e converter para radianos. Para realizar a conversão utilize a regra de três, considerando que PI radianos é igual a 180° e que a constante PI assume valor de 3.141592

- P16)** Sabendo que

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots = \sum_{i=0}^{\infty} (-1)^i \frac{1}{2i+1}$$

Escreva um programa, em linguagem C, que calcule uma aproximação a constante Π , somando os termos da série acima até que seu valor absoluto seja < 0.00001 . O algoritmo deve estar contido no arquivo "pi.c".

- P17)** Uma cidade foi atingida por uma moléstia epidêmica. Os setores de saúde calculam que o número de pessoas atingidas pela moléstia, depois de um certo tempo t , medidos em dias a partir do primeiro dia da epidemia, é, aproximadamente, dada por $f(t) = 64*t - (t^3)/3$. Escreva um programa, em linguagem C, que leia do *dispositivo padrão de entrada* (teclado) a data de início da epidemia, calcula os dias transcorridos (t) até a data de hoje. Calcule e mostre no *dispositivo padrão de saída* (tela de display) o número de pessoas atingidas no período, processando a função f . Nota: considere os anos bissextos para o cálculo dos dias transcorridos.

- P18)** Escreva um programa, em linguagem C, que leia um número inteiro positivo n e em seguida imprima n linhas do chamado Triângulo de Floyd. Para $n = 6$, temos:

```
01
02 03
04 05 06
07 08 09 10
11 12 13 14 15
16 17 18 19 20 21
```

O triângulo de Floyd tem diversas propriedades matemáticas interessantes. Os números do cateto esquerdo formam a sequência dos números poligonais centrais. Enquanto os da hipotenusa formam o conjunto de números triangulares. A soma dos números da linha n é equivalente a fórmula $n(n+1)/2$.