



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. O trabalho deve ser desenvolvido em grupo composto de 1 até 3 (um até três) estudantes e qualquer identificação de plágio sofrerá penalização;
3. O trabalho deve possuir uma capa, com identificação da disciplina, nomes dos estudantes, números de matrícula e e-mail.
4. Entrega dos programas gravados em CD/DVD, impreterivelmente, no dia **19/04/2016**.

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)** Durante o treino de uma corrida de automóveis, realizadas **N** voltas e foram anotados para um determinado piloto, na ordem, os tempos registrados em cada volta. Construir um algoritmo em *linguagem C* para ler os tempos de **N** voltas, calcular e imprimir:

- a) A volta de melhor tempo (menor);
- b) a volta em que o melhor tempo ocorreu;
- c) o tempo médio das **N** voltas;

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

- P3)** Você foi convocado pelo TSE para desenvolver o software da urna eletrônica que será utilizada no processo da eleição presidencial, onde três candidatos foram: **Fulano**, **Sicrano** e **Beltrano**. Os votos são informados através do código de inscrição. Os dados utilizados para o escrutínio obedecem a seguinte codificação: **Fulano**, código **1**; **Sicrano**, código **2**; **Beltrano**, código **3**; Voto nulo, código **4**; Voto em branco, código **5**. O software da urna eletrônica deve apresentar ao usuário um *menu* com as opções para que ele efetue a escolha, continua recolhendo votos até que um código de encerramento seja fornecido.

- a) Elabore o programa em linguagem C que efetua o escrutínio da eleição, onde se calcula e mostra as seguintes informações:

- Total de votos para cada candidato;
- Total de votos nulos;
- Total de votos em branco;

Como *finalizador* do conjunto de votos, tem-se o valor **99**.

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

- P4)** A *sequência de Fibonacci* se define os dois primeiros termos e cada termo seguinte é igual à soma dos dois termos imediatamente anteriores. O *n-ésimo* número da sequência de Fibonacci F_n é dado pela seguinte fórmula de recorrência descrita abaixo:

$$\begin{cases} F_0 = 0 \\ F_1 = 1 \\ F_n = F_{n-1} + F_{n-2} \quad \text{para } n \geq 2 \end{cases}$$

Os primeiros números de Fibonacci para $n = 0, 1, \dots$ são 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Escreva em linguagem C um programa que realiza a *Sequência de Fibonacci*, onde dado **n**, calcula **F_n**.

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

P5) 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".

P6) 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".

P7) Verificar se um dado número natural é ou não um Número Primo.

Teorema: Se um dado número natural **n** não possuir divisores $\leq \mathbf{raizq(n)}$, então **n** é um número Primo.

raizq(n) \rightarrow significa raiz quadrada de **n**

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

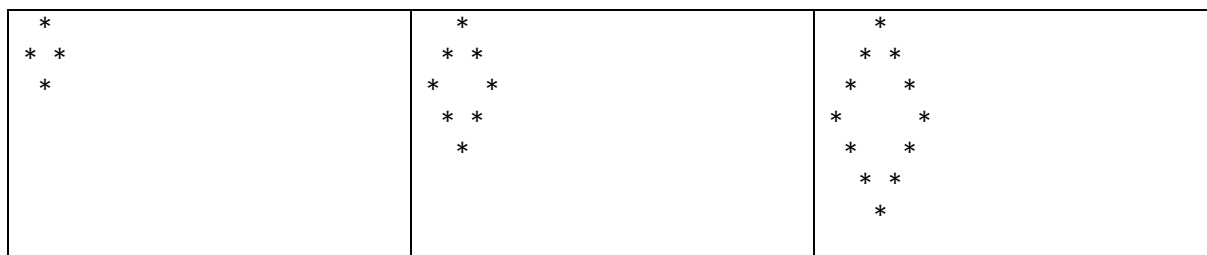
P8) Escreva um programa em linguagem C que construa uma pirâmide de números conforme ilustrado na figura abaixo. O programa deve ler do usuário a informação da altura da pirâmide **h**. Por exemplo, para uma altura **h=6** a pirâmide deve ser construída como na figura abaixo (automatize o algoritmo):

```

      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".

- P9)** Escreva um programa em linguagem C que escreva no dispositivo padrão de saída (tela) a figura de um losango, como mostrado na figura abaixo. A figura mostra um exemplo de losango para a altura de 4 asteriscos. A dimensão da altura deve ser informada pelo usuário e o algoritmo deve se adaptar as escolhas do usuário.



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

- P10)** 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:

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

P11) 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

P12) 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

P13) Sabendo que

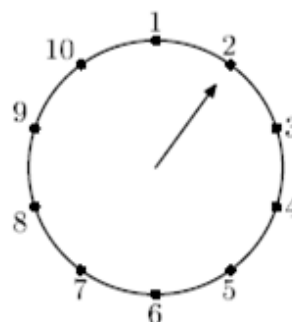
$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots = \sum_{i=0}^{\infty} (-1)^i \frac{1}{2i+1}$$

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".

P14) Considere o seguinte jogo: n pessoas (numeradas de 1 a n) são dispostas em círculo. O algoritmo deve eleger uma posição inicial em aleatório e remover as pessoas do círculo de forma alternada (pessoa-sim-pessoa-não) e o círculo aperta-se. O algoritmo termina quando restar apenas uma pessoa (sobrevivente).

Por exemplo, para $n = 10$ e posição inicial igual a 2, a ordem das eliminações é: 2, 4, 6, 8, 10, 3, 7, 1, 9 e 5 é o sobrevivente (ver figura).

Escreva em linguagem C um programa que leia n e escreva a ordem das eliminações e o sobrevivente de acordo com este algoritmo.



DICA: Represente o círculo de pessoas com uma variável indexada.

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

P15) Escreva um programa em linguagem C que capture uma string maior e uma menor. Depois verifica o número de vezes em que a menor aparece na maior. Um exemplo de execução é dado por:

Digite a string maior:
tres pratos de trigo para tres tigres tristes
Digite a string menor:
tres
Resultado: String menor aparece: 2 vezes
O algoritmo deve estar contido no arquivo "busca.c".

P16) Um *palíndromo* é uma palavra ou frase que tenha a propriedade de poder ser lida tanto da direita para a esquerda como da esquerda para a direita. Em um *palíndromo*, normalmente são desconsiderados os sinais ortográficos (diacríticos ou de pontuação), assim como os espaços entre palavras.

- Escreva em linguagem C um programa denominado *palíndromo.c* que leia do dispositivo padrão de entrada (*stdin*) uma cadeia de caracteres (string). A função deve desconsiderar os espaços ou sinais de pontuação. Se a string recebida for um palíndromo o programa deve escrever a mensagem "**eh palindromo**", caso contrário, escreve "**não eh palíndromo**".

Exemplos de palíndromos são:
Socorram-me, subi no onibus em Marrocos
Anotaram a data da maratona

P17) Escreva em linguagem de programação C um programa denominado *conta_palavras* que leia do dispositivo padrão de entrada (*stdin*) uma frase no formato de string. O programa deve calcular e mostrar na tela a quantidade de palavras contidas na frase lida.

Exemplo:
Frase: O unico homem que esta isento de erros, eh aquele que nao arrisca acertar.
Saída: A frase tem 14 palavras

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

P18) Escreva um programa em linguagem C que constrói uma matriz de sequência numérica em zig-zag (conforme o exemplo abaixo). O programa deve ler do dispositivo padrão de entrada o número de linha e colunas da matriz. O programa deve ler vários pares de linhas e colunas e encerrar as leituras quando o usuário digitar o par 0,0 para linha e coluna. Abaixo um exemplo de entrada e saída do programa:

Digite o numero de linhas e colunas da matriz: 3 4
A matriz zig-zag de tamanho 3x4 eh:
1 2 3 4
8 7 6 5
9 10 11 12

Digite o numero de linhas e colunas da matriz: 0 0
O algoritmo deve estar contido no arquivo "zigzag.c".

P19) Elabore um programa que carregue uma matriz 4x4 com números inteiros e verifique se essa matriz forma o chamado quadrado mágico. Um quadrado mágico é formado quando a soma dos elementos de cada linha é igual à soma dos elementos da coluna e igual a soma dos elementos das diagonais (principal e secundária).

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

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

P20) Um fabricante de automóveis coletou dados sobre níveis de ruído (medido em decibéis) produzidos por seis modelos de automóveis em sete (7) diferentes velocidades. Os dados foram registrados na tabela abaixo:

Carro	Velocidade (Km/h)						
	50	60	70	80	90	100	110
1	88	90	94	102	111	122	134
2	75	77	80	86	94	103	113
3	80	83	85	94	100	111	121
4	68	71	76	85	96	110	125
5	77	84	91	98	105	112	119
6	81	85	90	96	102	109	120

Escreva um programa em linguagem C que leia os dados da tabela e os apresente na tela de saída em um formato legal (parecido com o exemplo) e que calcule os níveis médio de ruído para cada automóvel, o nível de ruído médio para cada velocidade e o nível médio de ruído global.

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