Lista de Exercício de Programação

Instituto Federal Fluminense

Campus Campos Centro

Coordenação de Informática

Exercícios com Estruturas Sequenciais

1. Leia um valor de temperatura em graus Celsius e converta-o em graus Fahrenheit e Kelvin. A fórmula de conversão é:
   F = (9C + 160) / 5
   K = C + 273

2. Faça um programa que converta uma medida de metros para centímetros.

3. Faça um Programa que peça o comprimento a largura e a altura de uma caixa d'água retangular.
   Calcule e informe o volume da caixa.

4. Faça um algoritmo para ler valores para as variáveis A e B, e efetuar a troca dos valores de forma que, a variável A passe a possuir o valor da variável B e que a variável B passe a possuir o valor da variável A. Apresentar os valores trocados.

5. Prepare um algoritmo para calcular o espaço percorrido por um móvel em movimento retilíneo uniforme dada a seguinte fórmula: S=So+V*T.
   Inicialmente o algoritmo deverá perguntar ao usuário os valores do espaço inicial(So), da velocidade(V) e do tempo(T). Após a entrada dos devidos valores estes devem ser substituídos na fórmula, resultando assim no espaço percorrido(S) que deverá ser informado na tela do usuário.

6. Leia um número inteiro e imprima o seu antecedente e o seu sucessor.

7. Prepare um algoritmo para informar o total gasto em uma lavanderia. O algoritmo inicialmente deverá perguntar o total de camisas, o total de calças e o total de meias e informar o total gasto, levando em conta a seguinte tabela de preços:
   Camisas : 5.00 Calças :10.00 Meias :2.00

8. Implemente um programa que lê três valores e calcule a média ponderada para pesos 1, 2 e 3, respectivamente (multiplique cada nota pelo seu peso, some os produtos e divida o resultado pela soma dos pesos).

9. Uma criança quer saber qual é a soma de todas as idades que ela já teve. Elaborar programa que lê uma idade qualquer e responde rapidamente a essa pergunta (fórmula para calcular a soma dos N primeiros números inteiros: N (N+1)/2).

10. Dado o peso e a altura do usuário imprima seu Índice de Massa Corporal.
    IMC = peso / altura 2

11. Leia o salário mensal atual de um funcionário e o percentual de reajuste. Calcular e escrever o valor do novo salário.

12. Leia a idade de uma pessoa expressa em anos, meses e dias e escreva a idade dessa pessoa expressa apenas em dias. Considerar ano com 365 dias e mês com 30 dias.

13. Escrever um algoritmo que leia um valor e calcule qual o menor número possível de notas e moedas de 100, 50, 20, 10, 5, 2 e 1 em que o valor lido pode ser decomposto. Escrever o valor lido e a relação de notas necessárias.

Exercícios com Estruturas de Seleção

14. Faça um programa que peça um valor e mostre na tela se o valor é positivo ou negativo.

15. Dado o salário fixo, o valor das vendas efetuadas pelo vendedor de uma empresa e sabendo-se que ele recebe uma comissão de 3% sobre o total das vendas até R$ 1.500,00 e 5% sobre o que ultrapassar este valor, calcular e escrever o seu salário total.

16. Faça um Programa que verifique se uma letra digitada é "F" ou "M". Conforme a letra escrever: F - Feminino, M - Masculino, Sexo Inválido. Obs: O programa deve funcionar para letras maiúsculas e minúsculas.

17. Prepare um programa para perguntar dois números e informar qual deles é o maior.

18. Elabore um algoritmo que dada a idade de um nadador classifica-o em uma das seguintes categorias: infantil A = 5-7 anos; infantil B = 8-10 anos; juvenil A = 11-13 anos; juvenil B = 14-17 anos; adulto = maiores de 18 anos

19. Tendo como dados de entrada a altura o sexo e o peso de uma pessoa, construa um algoritmo que calcule seu peso ideal, utilizando as seguintes fórmulas:
    . Para homens: (72.7*h) - 58
    . Para mulheres: (62.1*h) - 44.7 (h = altura)
    Informe se o peso da pessoa está dentro, acima ou abaixo do peso (Considere a margem de erro de 1 Kg para mais ou para menos como estando no peso ideal).

20. Faça um Programa para uma loja de tintas. O programa deverá pedir o tamanho em metros quadrados da área a ser pintada. Considere que a cobertura da tinta é de 1 litro para cada 6 metros quadrados e que a tinta é vendida em latas de 18 litros, que custam R$ 80,00 ou em galões de 3,6 litros, que custam R$ 25,00.
    * Informe ao usuário as quantidades de tinta a serem compradas e os respectivos preços em 3 situações:
    * comprar apenas latas de 18 litros;
    * comprar apenas galões de 3,6 litros;
    * comprar latas e galões, a combinação de melhor resultado.

21. Faça um programa de menu que mostra na tela, sob o título de "Menu Principal", três opções: "1 - Fim", "2 - Cadastro" e "3 - Consulta", lê do teclado a opção desejada pelo usuário e mostra uma mensagem confirmando a opção escolhida ou uma mensagem de erro, se a opção for inválida.

22. Prepare um algoritmo para realizar o cálculo do preço de um serviço de Consultoria. Seu algoritmo deverá perguntar os seguintes dados e informar o valor total do serviço:
    Tipo de Serviço (Projeto ou Auditoria)
    N.º dias trabalhados
    N.º de viagens realizadas

Você deve usar a seguinte tabela para calcular o valor dos serviços:

Tipo de Serviço	Dia de Trabalho (R$)	Cada Viagem (R$)
Projeto	200,00	1000,00
Auditoria	100,00	1500,00

23. Prepare um algoritmo para realizar o cálculo do salário de uma pessoa. Seu algoritmo deverá perguntar os seguintes dados sobre o empregado:
    Cargo (Gerente, Supervisor, Servente)
    N.º Horas Extras trabalhadas
    N.º de Faltas
    N.º de Filhos

Você deve usar a seguinte tabela para calcular o valor dos salários:

Tipo de Empregado	Salário Base
Gerente	2000,00
Supervisor	900,00
Servente	300,00

Cada hora extra vale (Salário Base /240) * 2
Cada Falta custa: Salário Base /30
Cada Filho vale: 3% do salário base
INSS = 10% dos Proventos
Proventos = (salario base + horas extras + adicional por filhos)
Descontos = (Faltas + INSS)
Salário Líquido = Proventos - Descontos
Leia os dados e informe o total de Proventos, Descontos e o Salário Líquido.

24. Construa um algoritmo que receba a idade do usuário e verifique se ele tem mais de 21 anos.

25. Faça um Programa que peça para entrar com um ano com 4 dígitos e determine se o mesmo é ou não bissexto. Obs. Resposta em http://pt.wikipedia.org/wiki/Ano_bissexto

26. Faça um Programa que verifique se uma letra digitada é vogal ou consoante.

27. Criar um programa para calcular as raízes de uma equação de segundo grau. Inicialmente o programa deverá viabilizar a entrada de valores reais para as variáveis A, B e C. Em seguida deverá ser calculado o valor de DELTA. Caso o valor de DELTA seja menor do que 0 o programa deverá informar que não existem valores reais para as raízes, caso seja 0 deverá informar que existe somente uma raiz e o seu valor, caso contrário deverão ser informados os valores da raízes.
    Obs: Utilizar a função sqrt(valor) para calcular a raiz quadrada (biblioteca math.h);
    Utilizar a função pow(valor, potência) para elevar um número ao quadrado (biblioteca math.h);

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

    Fórmula para realizar os cálculos das raízes: Delta = $b^2 - 4ac$

28. Faça um programa para a leitura de duas notas parciais de um aluno. O programa deve calcular a média alcançada pelo aluno e apresentar:
* A mensagem "Aprovado", se a média alcançada for maior ou igual a sete;
* A mensagem "Reprovado", se a média for menor do que sete;
* A mensagem "Aprovado com Distinção", se a média for igual a dez.

29. Preparar um algoritmo para ler os comprimentos dos três lados de um triângulo (A,B e C) e determinar que tipo de triângulo temos, com base nos seguintes casos:

- Se A>=B+C nenhum triângulo é formado
- Se A2 = B2 + C2 um triângulo retângulo é formado
- Se A2 > B2 + C2 um triângulo obtusângulo é formado
- Se A2 < B2 + C2 um triângulo acutângulo é formado

30. Faça um programa que lê as duas notas parciais obtidas por um aluno numa disciplina ao longo de um semestre, e calcule a sua média. A atribuição de conceitos obedece à tabela abaixo:

Média de Aproveitamento Conceito
Entre 9.0 e 10.0 → A
Entre 7.5 e 9.0 → B
Entre 6.0 e 7.5 → C
Entre 4.0 e 6.0 → D
Entre 4.0 e zero → E
O algoritmo deve mostrar na tela as notas, a média, o conceito correspondente e a mensagem "APROVADO" se o conceito for A, B ou C ou "REPROVADO" se o conceito for D ou E.

31. Faça um programa que recebe o salário de um colaborador e o reajuste segundo o seguinte critério, baseado no salário atual:

* salários até R$ 280,00 (incluindo) : aumento de 20%
* salários entre R$ 280,00 e R$ 700,00 : aumento de 15%
* salários entre R$ 700,00 e R$ 1500,00 : aumento de 10%
* salários de R$ 1500,00 em diante : aumento de 5%

Após o aumento ser realizado, informe na tela:
* o salário antes do reajuste;
* o percentual de aumento aplicado;
* o valor do aumento;
* o novo salário, após o aumento.

32. Faça um Programa que leia um número inteiro menor que 1000 e imprima a quantidade de centenas, dezenas e unidades do mesmo.
* Observando os termos no plural a colocação do "e", da vírgula entre outros. Exemplo:
* 326 = 3 centenas, 2 dezenas e 6 unidades
* 12 = 1 dezena e 2 unidades Testar com: 326, 300, 100, 320, 310,305, 301, 101, 311, 111, 25, 20, 10, 21, 11, 1, 7 e 16

Exercícios com Estruturas de Repetição

33. Prepare um algoritmo para perguntar 10 números e informar a soma total destes números.

34. Refaça o algoritmo anterior para 100 números.

35. Faça um programa que escreva todos os números múltiplos de 7 entre 1 e N, sendo N um valor introduzido pelo utilizador. Por exemplos: 7, 14, 21, 28, 35.

36. Se tivermos uma lista dos números naturais menores do que 10 que são múltiplos de 3 ou 5 obtemos 3, 5, 6 e 9. A soma destes múltiplos é 23. Imprima a soma dos múltiplos de 3 ou 5 menores do que 1000.

37. Prepare um algoritmo para somar os números compreendidos entre 1 e 1000.

38. Prepare um algoritmo para calcular a soma dos números entre 1 e N inclusive. O valor de N deve ser perguntado no início do algoritmo.

39. Faça a multiplicação entre dois números usando somente soma.

40. Prepare um algoritmo para perguntar 40 números e informar o maior e o menor número lido.

41. Prepare um algoritmo para perguntar o nome e a altura de 20 pessoas e informar a média destas alturas, a altura da pessoa mais baixa e o nome da pessoa mais alta.

42. Prepare um algoritmo para perguntar o nome e o sexo de 10 pessoas e informar quantas são homens e quantas são mulheres.

43. Prepare um algoritmo para perguntar o nome, o sexo e a idade de 40 pessoas e informar:
    • a média de idades
    • a média de idades dos homens
    • o total de mulheres

44. Prepare um algoritmo para perguntar a um certo número de pessoas seu nome, sexo, peso e nacionalidade. Informe:

    • A média de peso destas pessoas
    • O nome da pessoa mais pesada
    • O nome da mulher brasileira mais leve

    Parar a execução do algoritmo quando o nome da pessoa for ASTROBALDO.

45. Prepare um algoritmo para controlar as informações sobre utilização de um banco eletrônico. Seu algoritmo deverá perguntar a 1000 clientes qual foi a operação que eles realizaram. Os tipos válidos são:
    - Retirada
    - Depósito
    - Extrato
    - Transferência

    Informe quantas operações foram efetuadas de cada tipo.

46. Faça um algoritmo para simular uma situação simples de depósito, retirada e consulta em um banco.
    O algoritmo inicialmente deverá mostrar um menu com as seguintes opções:
    1 - Depósito
    2 - Retirada
    3 - Saldo
    4 - Sair do algoritmo
    Se a escolha do usuário for depósito ou retirada, o algoritmo deverá pedir o valor da operação e atualizar automaticamente o valor existente na conta.
    O algoritmo deverá ser utilizado até que o usuário escolha a opção sair do algoritmo.

47. Faça um algoritmo para executar uma pesquisa entre um número de pessoas determinado pelo usuário. Após a entrada dos devidos dados informe:
    - O nome da pessoa mais nova e a sua idade
    - O nome da pessoa mais velha e a sua idade
    - A média das idades

48. Uma determinada companhia deseja obter o resultado de uma pesquisa relacionada a um novo produto. As respostas dos entrevistados devem ser apresentadas da seguinte maneira:
    - se o cliente gostou será digitada a letra S;
    - se o cliente não gostou será digitada será digitada a letra N;
    - se não for digitada nem a letra S nem a N, o algoritmo deverá ser finalizado;

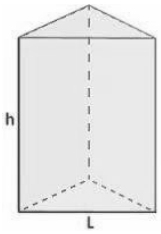
    Informe:
    Bom : se a maior parte dos entrevistados responder sim;
    Ruim : se a maior parte dos entrevistados responder não;
    Empate : caso ocorra empate;

49. Uma certa firma fez uma pesquisa de mercado para saber se as pessoas gostaram ou não de um novo produto lançado no mercado. Para isso, cadastrou o sexo do entrevistado e sua resposta (sim ou não). Sabendo-se que foram entrevistadas 2.000 pessoas, fazer um algoritmo que calcule e escreva:

    - o número de pessoas que responderam sim;
    - o número de pessoas que responderam não;
    - a porcentagem de pessoas do sexo feminino que responderam sim;
    - a porcentagem de pessoas do sexo masculino que responderam não;

50. Faça um algoritmo para calcular a área de figuras geométricas. Inicialmente o algoritmo deverá apresentar ao usuário um menu principal, onde será escolhida uma dentre as seguintes opções:
    1 - Calcular a área do quadrado
    2 - Calcular a área do retângulo
    3 - Calcular a área do triângulo
    4 - Calcular a área do círculo
    5 - Sair do algoritmo
    Após ser escolhida uma das figuras, o usuário deverá então entrar com as devidas medidas da mesma para que o cálculo possa ser realizado.
    Depois de informar o resultado do cálculo, o algoritmo deverá então voltar ao menu principal até que o usuário deseje finalizar o algoritmo.

51. Faça um algoritmo para trabalhar com prismas triangulares regulares de acordo com as fórmulas abaixo:
    área lateral (SL)= 3 * (área do retângulo)
    área da base (SB)= área do triângulo equilátero = lado^2 *
    área total (ST) = (2 * SB) + SL
    volume = SB * h



O usuário deverá entrar com valores para os lados e para as alturas dos prismas até que deseje finalizar o algoritmo (flag de saída a seu critério).

Após a entrada dos dados informe:

- O volume de cada prisma.
- O nº do prisma de maior área da base (usar contador para numerar os prismas durante a entrada de dados).
- O valor da menor área lateral.
- A média das áreas totais

1.

52. Fazer um programa que adivinhe um valor entre 0 e 100, no menor número de tentativas, recebendo do usuário a informação “menor”, “maior” e “acertou” (dica: a melhor estratégia é “chutar” o número mediano do intervalo considerado).

53. Neste problema você recriará um dos momentos verdadeiramente grandiosos da história, que é a clássica corrida entre a lebre e a tartaruga. Você usará a geração aleatória de números para desenvolver uma simulação desse memorável evento.
    Nossos competidores começam a corrida no "quadro1" de setenta quadrados. Cada quadrado representa uma posição possível ao longo do trajeto da corrida. A linha de chegada está no quadrado 70. O primeiro competidor a alcançar ou passar do quadrado 70 ganha uma cesta de cenouras e alface. O trajeto da corrida inclui uma subida pela encosta de uma montanha escorregadia, portanto, ocasionalmente, os competidores perdem terreno.
    Há um relógio que emite um tique por segundo. A cada tique do relógio, seu programa deve ajustar a posição dos animais de acordo com as seguintes regras:

Animal	Tipo de Movimento	Porcentagem de Tempo	Movimento real
Tartaruga	Movimento rápido	50%	3 quadrados para a direita
	Escorregão	20%	6 quadrados para a esquerda
	Movimento Lento	30%	1 quadrado para a direita
Lebre	Sono	20%	Nenhum movimento
	Salto grande	20%	9 quadrados para a direita
	Grande escorregão	10%	12 quadrados para a esquerda
	Salto pequeno	30%	1 quadrado para a direita
	Pequeno escorregão	20%	2 quadrados para a esquerda

Use variáveis para controlar as posições dos animais (i.e., os números das posições vão de 1 a 70). Cada animal inicia na posição 1 (i.e. "linha de partida"). Se um animal escorregar para antes do quadrado 1, leve o animal de volta ao quadrado 1.

Gere as porcentagens da tabela anterior produzindo um inteiro aleatório, i, no intervalo $1 \leq i \leq 10$. Para a tartaruga, realize um "movimento rápido" quando $1 \leq i \leq 5$, um " escorregão" quando $6 \leq i \leq 7$ ou um "movimento lento" quando $8 \leq i \leq 10$. Use uma técnica similar para mover a lebre.

Comece a corrida imprimindo:

BANG!!!!

E ELES PARTIRAM!!!!

Depois, para cada clique do relógio (i.e. a cada repetição do loop), imprima uma linha com 70 posições mostrando a letra **"T"** na posição da tartaruga e **"L"** na posição da lebre. Ocasionalmente, os competidores estarão no mesmo quadrado. Nesse caso, a tartaruga morde a lebre e seu programa deve imprimir **"OUCH!!!"** iniciando naquela posição. Todas as posições impressas diferentes das que levam o **"T"** , o **"L"** e a palavra **"OUCH!!!"** (no caso de os animais ocuparem o mesmo quadrado) devem estar em branco.

Depois de cada linha ser impressa, teste se algum dos animais alcançou ou passou do quadrado 70. Em caso positivo, imprima o vencedor e termine a simulação. Se a tartaruga vencer, imprima **TARTARUGA VENCEU!!! YAY!!!** Se a lebre vencer, imprima **LEBRE VENCEU!!! YUCH!!!** Se ambos os animais venceram no mesmo tique do relógio, você pode querer favorecer a tartaruga ("a parte mais fraca") ou pode desejar imprimir **Houve um Empate**. Se nenhum animal vencer, realize o loop novamente para simular o próximo tique do relógio.

Exercícios com String

54. Programa que lê duas palavras da entrada e, na saída, imprime a menor palavra.

55. Conta espaços e vogais. Dado uma string com uma frase informada pelo usuário (incluindo espaços em branco), conte:

    1. quantos espaços em branco existem na frase.
    2. quantas vezes aparecem as vogais a, e, i, o, u.

56. Tamanho de strings. Faça um programa que leia 2 strings e informe o conteúdo delas seguido do seu comprimento. Informe também se as duas strings possuem o mesmo comprimento e são iguais ou diferentes no conteúdo.

    Tamanho de "Brasil Hexa 2010": 16 caracteres
    Tamanho de "Brasil! Hexa 2010!": 18 caracteres
    As duas strings são de tamanhos diferentes.
    As duas strings possuem conteúdo diferente.

57. Nome na vertical. Faça um programa que solicite o nome do usuário e imprima-o na vertical.

    F
    U
    L
    A
    N
    O

58. Nome ao contrário em maiúsculas. Faça um programa que permita ao usuário digitar o seu nome e em seguida mostre o nome do usuário de trás para frente utilizando somente letras maiúsculas. Dica: lembre−se que ao informar o nome o usuário pode digitar letras maiúsculas ou minúsculas.

59. Faça um programa para após a entrada de dois nomes de no máximo 20 caracteres cada, implementar as seguintes funcionalidades:
    - apresentar os nomes de trás para frente.
    - informar se os nomes são iguais ou diferentes, porém, sem utilizar a função strcmp.
    - criar uma terceira variável de tamanho 40 e concatenar (junte) os nomes anteriormente digitados formando uma nova string. Fazer a impressão da string resultante.

60. Nome na vertical em escada. Modifique o programa anterior de forma a mostrar o nome em formato de escada.
    F
    FU
    FUL
    FULA
    FULAN
    FULANO

61. Nome na vertical em escada invertida. Altere o programa anterior de modo que a escada seja invertida.
    FULANO
    FULAN
    FULA
    FUL
    FU
    F

62. Construa uma função para receber uma string por parâmetro e retirar os caracteres de espaço em branco existentes na mesma.

63. Palíndromo. Um palíndromo é uma seqüência de caracteres cuja leitura é idêntica se feita da direita para esquerda ou vice−versa. Por exemplo: OSSO e OVO são palíndromos. Em textos mais complexos os espaços e pontuação são ignorados. A frase SUBI NO ONIBUS é o exemplo de uma frase palíndroma onde os espaços foram ignorados. Faça um programa que leia uma seqüência de caracteres, mostre−a e diga se é um palíndromo ou não.

Exercícios com Arrays (Vetores/Matrizes)

64. Prepare um programa para perguntar o nome e o sexo de 200 pessoas. Você deve listar primeiro o nome de todas as mulheres e depois o nome de todos os homens.

65. Prepare um programa para ler dois vetores, um com o preço e o outro com a quantidade vendida de 500 produtos. Seu programa deverá informar:
valor arrecadado com cada produto.
valor total arrecadado.

66. Prepare um programa para ler um vetor com 200 números. Depois desta leitura você deve calcular a média destes números. Seu programa deverá informar a posição e o valor de todos os números que ficaram acima desta média.

67. Ler um vetor com 50 números. Informar a posição em que está o maior número. Se o maior número existir mais de uma vez, informe todas as posições onde existir este número.

68. Elabore um programa que escreva qual o dia do ano, para uma dada data introduzida pelo usuário. O algoritmo recebe o dia, o mês e o ano, e depois mostra o dia do ano. Por exemplo, 23/02/2003 corresponde ao dia 54 do ano.

69. Prepare um programa para ler o nome e a idade de 30 pessoas. Após a leitura de dados o programa deverá informar o nome de todas as pessoas que ficarem acima da média de idades.

70. Prepare um programa para ler o nome de 50 pessoas e listá-los na ordem inversa de entrada.

71. Após ler dois vetores de inteiros A e B de tamanho N cada um:
-Armazenar em um terceiro vetor "Soma", a soma dos elementos do vetor A com os do vetor B (respeitando as mesmas posições) e escrever os elementos deste.
- Escrever a quantidade de vezes que A e B possuem os mesmos números nas mesmas posições.

72. Programa que leia do teclado um vetor de qualquer tamanho e inverta este vetor (sem usar funções prontas da linguagem).

73. Faça um programa que ordene um vetor, sem usar funções prontas da linguagem.

74. Faça um programa que leia um vetor de n elementos numéricos inteiros, calcule e mostre os números primos e suas respectivas posições.

75. Faça um programa que leia 20 elementos de um vetor A e construa um vetor B com os mesmos elementos de A, sendo que estes deverão estar invertidos, ou seja, o primeiro elemento de A passa a ser o último de B, o segundo de A passa a ser o penúltimo de B e assim por diante.

76. Faça um programa que leia dois vetores de 30 elementos e depois crie um terceiro que seja a junção dos dois primeiros (60 elementos).

77. Faça um programa para ler dois vetores de 25 elementos do tipo inteiro. Crie um terceiro vetor com tamanho 50 e inclua neste os elementos dos dois vetores anteriores de forma intercalada.

    Ex: vet1 1 3 5 7 ...
        vet2 2 4 6 8 ...
        vet3 1 2 3 4 5 6 7 8 ...

78. Faça um programa que leia um vetor de 15 elementos. Crie um segundo vetor onde cada elemento possua o fatorial do elemento do primeiro vetor.

79. Faça um programa para ler duas matrizes A e B, cada uma de duas dimensões com 5 linhas e 3 colunas. Construir uma matriz C de mesma dimensão, onde C é formada pela soma dos elementos da matriz A com os elementos da matriz B.

80. Faça um programa para ler dois vetores A e B com 7 elementos. construir uma matriz C de duas dimensões, onde a primeira coluna deverá ser formada pelos elementos do vetor A e a segunda pelos elementos do vetor B.

81. Faça um programa para ler dois vetores A e B com 12 elementos. Construir uma matriz C de duas dimensões, onde a primeira coluna da matriz deverá ser formada pelos elementos do vetor A multiplicados por 2 e a segunda coluna deverá ser formada pelos elementos do vetor B subtraídos de 5.

82. Faça um programa que simule uma pilha de livros. Um livro só poderá ser incluído no topo da pilha (respeitar o tamanho máximo do vetor - seu critério) , assim como sua retirada.
    Implemente os procedimentos de inserção, retirada e listagem da pilha.

83. Faça um programa que simule uma fila de um banco. Uma pessoa só pode entrar no final da fila, assim como sua saída só poderá ser feita no início da fila.
    Implemente os procedimentos de inclusão, saída e listagem da fila.
    Obs: Tamanho do vetor a seu critério.

84. Implemente um programa para inserção e remoção de elementos em um vetor ordenado, fazendo com que o mesmo permaneça neste estado.

85. Criar um programa de inserção e remoção para duas pilhas armazenadas em alocação sequencial que compartilham a memória de dimensão M.

86. Prepare um programa para controlar as informações de produção de uma fábrica com 100 produtos. Este controle é feito através de boletins que contém o número do produto (1 a 100) e a quantidade produzida. Após a leitura dos boletins, que deve ser finalizada quando o número do produto for igual a zero, o programa deverá informar:
    O total produzido de cada produto
    O produto mais produzido
    O produto menos produzido

87. Prepare um programa para controlar as informações sobre uma locadora de automóveis. Existem 10 tipos de carros que podem ser alugados. Sempre que um carro é alugado, são anotados o tipo (0.. 9) e o número de dias que o carro ficará com o cliente. Inicialmente o programa deverá perguntar o valor do aluguel, por dia, de cada um dos 20 tipos de carros.

Depois o programa deverá ler as informações sobre aluguel de carros (TIPO e DIAS) até que o tipo do carro seja colocado -1. Finalmente o programa deverá informar:
O valor arrecadado por cada tipo de carro.
O valor total arrecadado com o aluguel dos carros.

88. Prepare um programa para controlar a cobrança de um pedágio. Existem 10 tarifas diferentes que podem ser cobradas. Seu programa deverá inicialmente ler o preço de cada uma das 10 tarifas. Depois seu programa deverá ler o tipo de tarifa (1..10) até que seja digitado -1. Seu programa deverá informar:
    O total de pagamentos efetuados em cada tarifa.
    O total de pagamentos efetuados
    O valor total arrecadado em cada tarifa.
    O valor total arrecadado

89. Prepare um programa para controlar o estoque de uma loja. Inicialmente para cada um dos 20 produtos desta loja, o programa deverá perguntar seu nome e a quantidade inicial. Depois de ler estes dados iniciais o programa deverá ler várias fichas contendo:
    Número do produto (inteiro de 1 a 20)
    Tipo de transação ('C' ou 'V')
    Quantidade (real)
    Estas fichas podem ser tanto de compras (transação = 'C') quanto de vendas (transação = 'V'). Quando for compra o programa deverá incluir a quantidade no estoque. Se for venda deve retirar. Após leitura de todas as fichas que terminará com o número do produto = -1, o programa deverá informar para cada produto o seu nome e estoque atual.

90. Uma loja deseja controlar suas vendas de discos. Para isto, os discos foram divididos (0..4). Esta loja possui 15 vendedores que são conhecidos pelos nomes. Inicialmente, o programa deverá ler o preço unitário dos discos de cada um dos seis tipos e também o nome de cada um dos 15 vendedores. Quando uma venda é realizada, é preenchida uma ficha contendo o nome do vendedor e o tipo do disco que ele vendeu. Informe:
    A quantidade vendida de cada um dos seis tipos de disco.
    Total de discos vendidos por cada um dos vendedores.
    Valor total vendido por cada vendedor.

91. Escreva um algoritmo que leia uma lista de 40 elementos caractere que representa o gabarito de uma prova. A seguir, simulando a resposta de 50 alunos, gere de forma aleatória 50 vetores de resposta para cada um dos 50 alunos da turma. Mostre o nº de acertos dos alunos e a mensagem APROVADO, se a nota for maior ou igual a 60%; e mostre uma mensagem de REPROVADO, caso contrário.

92. Faça um programa que multiplique duas matrizes bi dimensionais.
    Matriz A tamanho 3 x 2;
    Matriz B tamanho 2 x 2;

Exercícios com Funções

93. Faça um programa para imprimir:

```
1
2  2
3  3  3
.....
n  n  n  n  n  n ... n
```

para um n informado pelo usuário. Use uma função que receba um valor n inteiro e imprima até a n-ésima linha.

94. Faça um programa para imprimir:

```
1
1  2
1  2  3
.....
1  2  3  ...  n
```

para um n informado pelo usuário. Use uma função que receba um valor n inteiro imprima até a n- ésima linha.

95. Faça um programa, com uma função que necessite de três argumentos, e que forneça a soma desses três argumentos.

96. Faça um programa, com uma função que necessite de um argumento. A função retorna o valor de caractere ‘P’, se seu argumento for positivo, e ‘N’, se seu argumento for zero ou negativo.

97. Faça um programa com uma função chamada somaImposto. A função possui dois parâmetros formais: taxaImposto, que é a quantia de imposto sobre vendas expressa em porcentagem e custo, que é o custo de um item antes do imposto. A função “altera” o valor de custo para incluir o imposto sobre vendas.

98. Faça um programa que converta da notação de 24 horas para a notação de 12 horas. Por exemplo, o programa deve converter 14:25 em 2:25. A entrada é dada em dois inteiros. Fazer uma função para a conversão do valor. Inclua um loop que permita que o usuário repita esse cálculo para novos valores de entrada todas as vezes que desejar.

99. Faça um programa que use a função valorPagamento para determinar o valor a ser pago por uma prestação de uma conta. O programa deverá solicitar ao usuário o valor da prestação e o número de dias em atraso e passar estes valores para a função valorPagamento, que calculará o valor a ser pago e devolverá este valor ao programa que a chamou. O programa deverá então exibir o valor a ser pago na tela. Após a execução o programa deverá voltar a pedir outro valor de prestação e assim continuar até que seja informado um valor igual a zero para a prestação. Neste momento o programa deverá ser encerrado, exibindo o relatório do dia, que conterá a quantidade e o valor total de prestações pagas no dia. O cálculo do valor a ser pago é feito da seguinte forma. Para pagamentos sem atraso, cobrar o valor da prestação. Quando houver atraso, cobrar 3% de multa, mais 0,1% de juros por dia de atraso.

100. Faça uma função que informe a quantidade de dígitos de um determinado número inteiro informado.

101. Reverso do número. Faça uma função que retorne o reverso de um número inteiro informado. Por exemplo: 127 -> 721.

102. Data com mês por extenso. Construa uma função que receba uma data no formato DD/MM/AAAA e devolva uma string no formato D de mesPorExtenso de AAAA. Opcionalmente, valide a data e retorne NULL caso a data seja inválida.

103. Desenha moldura. Construa uma função que desenhe um retângulo usando os caracteres '+' , '−' e '| '. Esta função deve receber dois parâmetros, linhas e colunas, sendo que o valor por omissão é o valor mínimo igual a 1 e o valor máximo é 20. Se valores fora da faixa forem informados, eles devem ser modificados para valores dentro da faixa de forma elegante.

104. Faça uma função que retorne a multiplicação entre dois números usando somente soma.

105. Faça uma função que receba um número e diga se é um número perfeito. Um número é perfeito se a soma de seus divisores é igual ao próprio número. Ex: 1+ 2 + 3 = 6, 6 é um número perfeito.

106. Leia as coordenadas cartesianas de 2 pontos A=(xA,yA), B=(xB,yB) e , usando funções, calcule:
(a) a distância entre os pontos.

$$d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

(b) o coeficiente angular da reta AB.
Ca= (y2 – y1) / (x2 – x1)

107. Usando funções, calcule o n-ésimo termo da sequência de Fibonacci, {1, 1, 2, 3, 5, 8, ...}:
=> de forma iterativa.
=> de forma recursiva.

$$F(n) = \begin{cases} 0 & \text{se } n = 0 \\ 1 & \text{se } n = 1 \\ F(n-1) + F(n-2) & \text{se } n > 1 \end{cases}$$

108. Implemente uma função que calcule a área da superfície e o volume de uma esfera de raio r. A área da superfície e o volume são dados, respectivamente, por $4\pi r^2$ e $4\pi r^3 / 3$. Essa função deve obedecer o seguinte protótipo: void calc_esfera (float r, float* area,float* volume).

Exercícios com Funções, Vetores e Strings

109. Implemente uma função que receba como parâmetro um vetor de números reais (vet) de tamanho n e retorne quantos números negativos estão armazenados nesse vetor. Essa função deve obedecer ao protótipo: int negativos (int n, float* vet);

110. Implemente uma função que receba como parâmetro um vetor de números inteiros (vet) de tamanho n e inverta a ordem dos elementos armazenados nesse vetor. Essa função deve obedecer ao protótipo: void inverte (int n, int* vet);

111. Construa uma função que receba uma mensagem, seu tamanho e um caracter e retire todas as ocorrências desse caracter na mensagem colocando * em seu lugar. A função deve retornar o total de caracteres retirados.

112. Embaralha palavra. Construa uma função que receba uma string como parâmetro e devolva outra string com os carateres embaralhados. Por exemplo: se função receber a palavra python, pode retornar npthyo, ophtyn ou qualquer outra combinação possível, de forma aleatória. Padronize em sua função que todos os caracteres serão devolvidos em caixa alta ou caixa baixa, independentemente de como foram digitados.

113. Implemente uma função que receba como parâmetros uma cadeia de caracteres s e um caractere c, e retorne o índice da última ocorrência do caractere c em s. Por exemplo, se forem passados para essa função a cadeia “Rio de Janeiro” e o caractere ‘i’, a função deve retornar o valor 11. Caso não haja ocorrências do caractere procurado, a função deve retornar -1. Essa função deve ter o seguinte protótipo: int ultima_ocorrencia (char* s, char c);

114. Faça uma função para receber uma lista de nomes e um nome a ser procurado. Caso o nome esteja na lista a função deverá retornar a posição do mesmo na lista, caso contrário deverá retornar -1.

115. Faça uma função que receba uma string e retorne o seu tamanho. Obs. Não utilizar funções prontas da linguagem.

116. Faça uma função que receba duas strings e retorne 0 se estas forem iguais e 1 se forem diferentes. Obs. Não utilizar funções prontas da linguagem, deverão ser comparados caractere a caractere.

117. Crie um programa que para cadastrar 10 pessoas em uma lista . Faça um laço que executa um bloco de código 10 vezes, solicita um nome completo ao usuário e o insere na lista.

Crie uma outra lista de tamanho 10. Preencha cada posição desta nova lista com:
- O último nome da pessoa
- Os nomes do meio abreviados
- O primeiro nome da pessoa

O nome completo abreviado deve ser colocado em uma posição da lista.
Exemplo:
Um lista de 3 posições:
0. João Silva Souza

1. Manoel Medeiros Mendes Cunha
2. Sebastião Rosa

Deve ser copiado para uma nova lista de 3 linhas:
0. Souza S João
1. Cunha M M Manoel
2. Rosa Sebastião

Imprima o conteúdo da lista.

118. Faça um programa para controle de uma locadora. Implemente os seguintes procedimentos:

Aquisição de um filme. A locadora, devido a seu espaço físico, se limita a ter no máximo 2000 filmes. Faça uma implementação de inclusão que atenda a este limite. Sugestão: Utilize uma variável para controlar o número de filmes existentes na locadora, a cada título novo adquirido some um a esta variável.
O registro de filmes deve conter os seguintes campos:
Título
Sinopse
Série (ouro, prata ou bronze)
Ator Principal
Alugado

-Remoção de Filme. A locadora poderá se desfazer de um título por motivos como danificação, roubo, etc. Implemente um procedimento que após a entrada do título do filme diminua um daquela variável de controle utilizada no procedimento de inclusão e logo após elimine o registro (como se trata de um vetor, a posição em que o registro se encontra não será realmente excluída, mas terão que entrar aí dados válidos de outros filmes).
-Aluguel de filme. O usuário deverá perguntar o nome(título) do filme e então se este não estiver alugado , marcá-lo como tal.
-Devolução de um filme. O usuário deverá perguntar o nome(título) do filme e então se este estiver alugado , marcá-lo como disponível.
-Listagem de filmes por ator. Pedir o nome do Ator e listar o nome de todos os filmes nos quais este ator trabalhar.
-Listagem de todos os dados dos filmes disponíveis naquele momento.
Informação do status de um filme. Entrar com o nome do filme, o programa deverá informar se este está alugado ou não.

Exercícios com Ponteiros e Alocação Dinâmica de Memória

Fonte principais: http://www.ime.usp.br/~pf/algoritmos/index.html
http://facoprograma.tumblr.com/exe_lista_ligada

119. Declare uma variável inteira n e um ponteiro ptrn para essa variável. Atribua um valor para n. A seguir, imprima o valor de n a partir de ptrn. Imprima também o endereço de n e de ptrn.

120. A partir do exercício anterior, incremente o valor de n usando apenas o ponteiro ptrn. Imprima o novo valor de n para confirmar se está correto.

121. Declare duas variáveis inteiras m e n e dois ponteiros para elas. Atribua valores para as variáveis. Descubra qual o maior valor, acessando-as somente através dos ponteiros. Imprima esse valor.

122. Descubra (e imprima) o valor da maior variável do exercício anterior, através da função maior, cujos parâmetros devem ser m e n.

123. O que aconteceria se você incrementasse os valores dos parâmetros, dentro da função? Quais seriam seus valores dentro da função? E fora dela? Teste e explique por quê.

124. Descubra (e imprima) o valor da maior variável do exercício anterior, através da função maior, cujos parâmetros devem ser as variáveis ponteiros para m e n.

125. O que aconteceria se você incrementasse os valores dos parâmetros, dentro da função? Quais seriam seus valores dentro da função? E fora dela? Teste e explique por quê.

126. Implemente uma função que receba como parâmetro um vetor v de n números inteiros e retorne um novo vetor w, alocado dinamicamente, cujos elementos são definidos pelas fórmulas:
w[0] = v[0]
w[i] = v[i] + w[i] − 1 , 0 < i < n
Essa função não deve alterar o conteúdo do vetor original v e seu protótipo deve ser:
int* somatorios (int n, int* v);

127. Implemente uma função que receba um vetor de inteiros (vet) de tamanho n. Essa função deve alocar dinâmicamente um outro vetor também de tamanho n que contenha os endereços dos valores do vetor de inteiros de forma ordenada crescente, ficando a primeira posição do vetor de ponteiros o endereço do menor valor até a última posição, que conterá o endereço do maior valor. Essa função deve obedecer ao protótipo: int**inverte2 (int n, int* vet);

Exercícios com Listas, Filas e Pilhas

Fonte principais: http://www.ime.usp.br/~pf/algoritmos/index.html
http://facoprograma.tumblr.com/exe_lista_ligada

128. Implementar um programa para manipulação de uma lista ordenada usando alocação dinâmica. Operações que devem ser contempladas:
- Inicializar lista
- Verificar lista vazia
- Inserir elemento
- Remover elemento
- Tamanho: retorna o número de elementos da lista
- Média: retorna a média aritmética simples dos elementos da lista
- Desalocar todos os nós da lista encadeada

129. A profundidade de uma célula c em uma lista encadeada é o número de passos do único caminho que vai da primeira célula da lista até c. Escreva uma função que calcule a profundidade de uma dada célula.

130. Escreva uma função que verifique se uma lista encadeada que contém números inteiros está em ordem crescente.

131. Dada uma lista encadeada de números inteiros cujo tipo que representa um nó da lista é dado por:

```
struct lista
{
        int info;
        struct lista *prox;
};
typedef struct lista Lista;
```

Implemente uma função que receba uma lista encadeada do tipo Lista e retorne um vetor de números inteiros, alocado dinamicamente, contendo as informações presentes na mesma ordem que na lista, e o número de elementos neste vetor.
Se a lista for vazia, o vetor retornado deverá ser NULO (NULL) e o número de elementos igual a 0.
O protótipo da função é dado por: int* Copia_Lista(Lista* L, int* n);

132. Escreva uma função que faça uma busca em uma lista encadeada crescente e retorne o endereço da estrutura encontrada.

133. Escreva uma função que encontre uma célula com conteúdo mínimo.

134. Escreva uma função que verifique se duas listas encadeadas são iguais, ou melhor, se têm o mesmo conteúdo.

135. Escreva uma função que faça uma cópia de uma lista encadeada.

136. Escreva uma função que concatene duas listas encadeadas (isto é, "engate" a segunda no fim da primeira).

137. Escreva uma função que inverta a ordem das células de uma lista encadeada (a primeira passa a ser a última, a segunda passa a ser a penúltima etc.). Faça isso sem usar espaço auxiliar, apenas alterando ponteiros.

138. Escreva uma função para remover de uma lista encadeada todas as células que contêm o valor x.

139. Escreva uma função que remova a k-ésima célula de uma lista encadeada.

140. Escreva uma função que tenha como parâmetros ponteiros para duas listas. A função deve retornar um ponteiro para uma terceira lista, que é o resultado da união das duas (isto é, os elementos repetidos devem ser considerados apenas uma vez).

141. Escreva uma função que tenha como parâmetros ponteiros para duas listas. A função deve retornar um ponteiro para uma terceira lista, que é o resultado da intersecção das duas (isto é, deve retornar apenas os elementos presentes em ambas).

142. Escreva uma função que receba um texto e imprima uma relação de todas as palavras que ocorrem no texto juntamente com o número de ocorrências de cada palavra. Use uma lista encadeada para armazenar as palavras.

143. Desenvolva um algoritmo para testar se duas pilhas P1 e P2 são iguais. Duas pilhas são guais se possuem os mesmos elementos, na mesma ordem.
Sua função deverá retornar 1 para p1 == p2 e 0 para p1 != p2.

144. Faça um programa para inverter a ordem dos elementos existentes em uma string de tamanho máximo 20. Utilize, obrigatoriamente, uma pilha como memória temporária para armazenar a string. Imprima a string após a inversão.

145. Construa um programa, que use a estrutura pilha e verifique se o número de abre parênteses é igual ao número de fecha parênteses em uma expressão matemática.

146. Implemente uma função que receba três filas, f, impares e pares (todas já inicializadas), e separe todos os valores guardados em f de tal forma que os valores pares são colocados na fila pares e os valores ímpares na impares. Ao final da execução desta função, f permanecerá com todos os seus elementos. O protótipo desta função é o seguinte: void separa_filas(TFila* f, TFila* par,TFila* impar).

147. O sistema de controle de fila de um banco opera da seguinte forma. O cliente que chega à agência retira uma senha que pode ter prioridade preferencial (para gestantes, portadores de necessidades especiais e maiores de 65 anos) ou não preferencial. A agência dispõe de um caixa preferencial e diversos caixas não preferenciais. Ao decidir qual o próximo cliente a ser atendido, são consideradas duas políticas: i. O caixa preferencial seleciona o primeiro cliente preferencial da fila. Não havendo clientes preferenciais, é selecionado o primeiro cliente da fila; ii. O caixa não preferencial respeita a ordem de chegada e seleciona sempre o primeiro cliente que está aguardando na fila, preferencial ou não. Para armazenar as informações da fila, o sistema utiliza uma lista duplamente encadeada, em que cada nó é do tipo Cliente, definido a partir da estrutura cliente, descrita a seguir:

```
struct cliente
{
        int senha;
```

```
        char prioridade;
        struct cliente *proximo;
};
typedef struct cliente Cliente;
```

Nessa estrutura, o campo senha guarda o número da senha atribuída a um cliente, o campo prioridade é um caractere que pode ter os valores 'P' ou 'N', indicando que o cliente é preferencial ou não preferencial, e o campo proximo aponta para o próximo nó da lista.

a) Escreva uma função em C que receba como parâmetros o ponteiro lst para a lista de clientes, o inteiro senha e o caractere prio. Os dois últimos valores são usados para preencher os campos senha e prio de uma variável Cliente que deve ser alocada dinamicamente e inserida no final da lista. A função, que deve retornar o ponteiro para o início da lista atualizado, tem o seguinte protótipo: Cliente* Enfileira(Cliente* lst, int senha, char prio);

b) Escreva uma função em C que receba como parâmetros o ponteiro lst para a lista de clientes e o caractere prt --- que tem o valor 'P' para indicar que a seleção deve ser feita para um caixa preferencial, ou 'N', para um caixa não preferencial --- e retorne a senha do cliente selecionado para atendimento, segundo as políticas apresentadas. Ou seja, se prt = 'P' a função deve retornar o primeiro cliente com campo prio = 'P' encontrado, ou, se não houver nenhum, o primeiro cliente da lista. Se prt = 'N' a função deve retornar o primeiro cliente da lista. O respectivo nó deve ser retirado da lista e liberado. Se a lista estiver vazia a função deve retornar -1. A função tem o seguinte protótipo: int Seleciona(Cliente** lst, char prt);

148. Descreva, em linguagem C, a estrutura de uma célula de uma lista duplamente encadeada.

149. Escreva uma função que remova de uma lista duplamente encadeada a célula apontada por p.

Exercícios com árvore binária de busca

150. Suponha que as chaves 50 30 70 20 40 60 80 15 25 35 45 36 são inseridas, nesta ordem, numa árvore de busca inicialmente vazia. Desenhe a árvore resultante.

151. Construir uma função para inserir um elemento em uma árvore binária de busca. Fazer uma versão recursiva e uma iterativa.

152. Construir uma função para percorrer a árvore, imprimindo os valores dos nós, segundo a estratégia em ordem.

153. Construir uma função para percorrer a árvore, imprimindo os valores dos nós, segundo a estratégia pré-ordem.

154. Construir uma função para percorrer a árvore, imprimindo os valores dos nós, segundo a estratégia pós-ordem.

155. Construir uma função para verificar se um certo valor n está presente na árvore.

156. Construir uma função para retornar o maior valor presente em uma árvore.

157. Construir uma função para retornar o menor valor presente em uma árvore.

158. Construir uma função para retornar a média dos valores presentes em uma árvore.

159. Construir uma função para retornar o número de NULLs presentes em uma árvore.

160. Construir uma função para retornar a soma dos valores dos nós.

161. Construir uma função para retornar o número de nós cujos valores são múltiplos de três.

162. Construir uma função para retornar o número de nós de uma árvore.

163. Construir uma função para retornar o número de folhas de uma árvore.

164. Construir uma função para retornar a altura de uma árvore.

165. Construir uma função para verificar se uma árvore é estritamente binária (uma árvore é estritamente binária caso todos os seus nós possuam 2 filhos ou sejam folhas).

166. Construir uma função para criar uma cópia de uma árvore.

167. Suponha que as chaves 50 30 70 20 40 60 80 15 25 35 45 36 são inseridas, nesta ordem, numa árvore de busca inicialmente vazia. Desenhe a árvore resultante. Em seguida remova o nó que contém 30.

168. Escreva uma versão recursiva da função que remove um nó de uma árvore de busca.

Escreva uma função que decida se uma dada árvore binária é ou não é de busca.

169.

Exercícios de Orientação a Objeto

170. Crie uma classe que modele uma bola e permita trocar e consultar a cor da bola.

171. Crie uma classe que modele um quadrado e permita definir, alterar e consultar o tamanho dos lados e obter a área.

172. Crie uma classe que modele um retângulo e permita definir, alterar e consultar os valores dos lados, obter a área e obter o perímetro.

173. Crie uma classe que modele uma pessoa e permita definir e obter a idade, peso e altura da pessoa e que permita fazer a pessoa envelhecer, engordar e emagrecer. A cada ano que a pessoa envelhece, sendo a idade dela menor que 21 anos, ela deve crescer 1,5 cm.

174. Crie uma classe que modele uma televisão e que permita ligar e desligar a TV, e alterar e consultar o número do canal e o volume. O número do canal e o volume somente podem ser alterados dentro de uma faixa válida.

175. Crie uma classe que modele uma conta corrente e que permita definir e consultar o numero da conta e o nome correntista, consultar o saldo e fazer depósitos e saques.

176. Crie uma classe que modele uma bomba de combustível e permita definir a capacidade da bomba, o preço do combustível por litro e permita encher a bomba (sempre encher completamente) e consultar sua quantidade de combustível. É necessário também que seja permitido abastecer por valor (retornando a quantidade de combustível abastecida), abastecer por quantidade de litros (retornando o valor a ser pago). Deve ser permitido também alterar o valor do preço por litro. Deve ser tratada a situação de não haver combustível suficiente na bomba para um determinado abastecimento.

177. Crie uma classe que modele um retângulo e permita definir a altura e a largura. A classe também deve permitir definir e alterar o ponto que representa o centro do retângulo e permitir que sejam consultados os pontos dos quatro vértices. Um ponto deve ser modelado por outra classe, e deve permitir definir e consultar valores x e y do ponto. O objeto também deve informar sua área, seu perímetro e se é um quadrado.

178. Crie uma classe chamada Carnívoro que permita que objetos desta classe se alimentem de qualquer coisa (números, booleanos, caracteres, strings ou qualquer outro objeto). O conteúdo do estômago do carnívoro deve poder ser consultado e também deve ser possível digerir os alimentos. A cada digestão o alimento mais antigo deve ser eliminado.

179. Crie um objeto número que seja capaz de informar:
a) os seus fatores primos;
b) seu fatorial;
c) o máximo divisor comum (MDC) entre ele mesmo e outro número;
d) o mínimo múltiplo comum (MMC) entre ele mesmo e outro número.

180. Crie uma classe que modele um número complexo para realizar aritmética com números complexos.

Os números complexos têm a forma:
parteReal + parteImaginaria * i, onde i é a raiz quadrada de -1.

181. Utilize variáveis reais para representar as partes real e imaginária do número. A presença das partes real e imaginária é condição necessária para a criação do objeto. Um número complexo deve ser capaz de realizar:

a) adição com outro número complexo: as partes reais são somadas de um lado e as partes imaginárias são somadas de outro;

b) subtração com outro número complexo: a parte real do operando direito é subtraída da parte real do operando esquerdo e a parte imaginária do operando direito é subtraída da parte imaginária do operando esquerdo;

c) multiplicação com outro número complexo;

d) divisão com outro número complexo;

e) fornecer uma representação em String do número na forma a + bi, onde a é a parte real e b é a parte imaginária.

As fórmulas podem ser encontradas em http://en.wikipedia.org/wiki/Complex_number

182. Crie uma classe que modele um número racional para realizar aritmética com frações. Um número racional é aquele que pode ser escrito na forma x/y. O valor do numerador e do denominador são obrigatórios. O construtor deve armazenar a fração na forma reduzida (isto é, a fração 2/4 seria armazenada no objeto como 1 no numerador e 2 no denominador). Um número racional também deve ser capaz de realizar:

a) adição com outro número racional;

b) subtração com outro número racional;

c) multiplicação com outro número racional;

d) divisão com outro número racional;

e) fornecimento de uma representação em String do número, na forma a/b, onde a é o numerador e b é o denominador.

f) fornecimento de uma representação em String do número em forma de ponto flutuante. Forneça também opção de formatação com o número de casas decimais.

183. Crie a classe Carta, que possui um nome e um naipe. Crie agora uma classe Baralho, que possui 52 cartas. No construtor de Baralho, inicialize as 52 cartas (juntamente com os 4 coringas, que devem se chamar “coringa” e o naipe é “coringa”). Escreva os seguintes métodos:

a) embaralha( ) – usando o método Math.random dá para trocar as cartas dentro do baralho, misturando-as (sorteie duas posições e troque-as. Repita este processo diversas vezes).

b) daCarta( ) – devolve uma carta, retirada do topo do baralho, se não estiver vazio, ou null.

c) temCarta( ) – verifica se tem carta no baralho, devolvendo true ou false. A classe Baralho, assim construída, será útil para programar diversos jogos de cartas, através de novas classes que os implementem.

d) imprimeBaralho( ) –imprime as cartas para verificar como estão dispostas (se estão embaralhadas por exemplo).

184. Faça um pequeno programa para implementar um motor de corrente alternada para ventiladores. O motor deverá ter as seguintes características: potência, rotações por

minuto(rpm) e tempo de vida útil. Crie uma classe motor que receba os parâmetros do motor, faça a inicialização do mesmo e controle a sua vida útil.

A vida útil do motor é controlada por um método “utilizar motor” que recebe o tempo que o motor será utilizado naquele momento e subtrai do tempo de vida útil.
Neste exemplo não existe problema do motor ficar com tempo de vida útil menor do que zero.
Obs1:A potência do motor e o rpm não podem ser alterados, somente consultados.
Obs2:Deverão ser criados testes unitários para todas as funcionalidades.
Obs3:Todos os detalhes de implementação necessários para o funcionamento do programa deverão ser criados!

185. Escreva um programa com o propósito de representar um ventilador do mundo real. Uma classe Ventilador deverá possuir os seguintes campos de dados e métodos:

a) Um campo de dados privado “motor” que seja formado pela instanciação de um motor da questão anterior.
b) Um campo de dados privado chamado velocidade, que define a velocidade atual do ventilador, cujos valores possíveis são Baixa, Média e Alta(o padrão é BAIXA).
c) Um campo de dados privado booleano que mostra se o ventilador está ligado ou desligado. O padrão é desligado, ou seja, False.
d) Um campo de dados privado que define a cor do ventilador.
e) Um método para ligar o ventilador. Para poupar o motor, um ventilador somente poderá ser ligado se o tempo de vida útil do motor dele for maior do que 0, senão uma exceção deverá ser levantada por este ventilador. Toda vez que um motor é ligado, o método de ligar deverá receber um parâmetro definindo o tempo estimado que o ventilador será utilizado naquele momento. Este valor deverá ser repassado ao motor que deverá ter a sua vida útil reduzida.
f) Demais métodos para obter e definir valores para os campos de dados privados e desligar o ventilador.
g) Um construtor com argumento cor, que cria um novo ventilador com os seus respectivos padrões de valores e a cor definida no parâmetro.
h) Um método que cria uma representação para um objeto ventilador, que deve retorna uma descrição textual do ventilador. A string retornada mostrará todos os dados do ventilador: cor, velocidade e se está ligado ou desligado. O método deverá sobrescrever o método padrão de representação do objeto.
i) Deverão ser criados testes unitários para todas as funcionalidades, inclusive para a exceção levantada.

186. Crie uma classe para instanciar objetos “números malucos”. Os objetos instanciados com esta classe deverão possuir os seguintes comportamentos:
- Quando somados dois “números malucos” o resultado deverá ser um terceiro objeto “número maluco” contendo o valor correspondente a multiplicação de dois números;
- Quando subtraídos dois números malucos, o resultado deverá ser ser um terceiro objeto “número maluco” contendo o valor correspondente a soma de dois números;
- Quando solicitada a impressão de um número maluco, o resultado deverá ser igual a: “NumeroMaluco(<valor_inteiro_do_numero>)”;
Crie também uma classe de teste que verifique se as funcionalidades da classe NumeroMaluco estão Ok.

187. Uma empresa que trabalha com sistema de franquias possui um grande números de franqueados. Cada franquia possui razão social, cnpj e um quadro de funcionários. O cnpj dos franqueados deverá ser validado* e deverá ser um objeto. Cada funcionário das franquias possue nome, data de nascimento, cpf e um controle se o mesmo está ativo ou não.

O seu programa deverá ser capaz de instanciar objetos funcionário e armazená-los nas franquias. Objetos funcionário deverão ser capazes de informar as suas idades (pode ser baseado somente no ano de nascimento). O franqueado deverá ter duas funcionalidades: dispensar funcionários e listar funcionários ativos.

Deverão ser criados testes para validar todas as classes métodos utilizados na implementação.

188. Uma escola está organizando um evento e gostaria de controlar as inscrições e cadastros dos alunos nos seminários organizados. Para tanto, faz-se necessário possibilitar a inclusão de seminários em um sistema contendo informações como código do seminário, descrição e número máximo de alunos participantes. É necessário ainda ter uma lista de espera, onde um aluno poderá ser incluído e, caso algum outro aluno desista deverá haver uma inclusão automática do primeiro que estiver na fila de espera. Para cada aluno é importante saber o nome, a matrícula, a data de nascimento e os seminários que o mesmo participou.

Obs1: é obrigatório a sobrecarga da igualdade entre objetos para que esta possa ser utilizada nos testes;

Obs2: testes deverão ser implementados para todas as funcionalidades do sistema;

Obs3: todos os imports e detalhes de implementação deverão ser implementados;

189. Faça um programa para instanciar objetos casa. Uma casa possuirá cor, que poderá ser alterada pintando-a. Estas casas, que serão por padrão retangulares, possuirão comprimento e largura, e utilizando destes atributos a área que a mesma ocupa poderá ser informada. As casas também deverão conter objetos porta, as quais possuirão cor e também poderão ser pintadas. Estas portas poderão estar abertas ou fechadas, seus estados poderão ser alterados e a consulta se uma porta está aberta deverá estar disponível. Na casa poderão ser adicionadas quantas portas forem desejadas. Será necessário consultar quantas portas existem na casa e o número de portas abertas. É necessário ainda fazer uma listagem dividindo as portas da casa em grupos por igualdade, sendo que para uma porta ser igual a outra, suas dimensões e cores precisarão ser iguais.

Implemente as classes e os testes para o contexto apresentado.

Obs:Todos os detalhes de implementação deverão estar presentes.

Mini-Mundos

190. Uma loja de venda de Eletrodomésticos quer automatizar o seu controle de compra e troca de aparelhos por parte de seus clientes. Todo aparelho vendido possui garantia de 1 ano, a partir da data de venda. Isto significa que qualquer troca só poderá ser realizada dentro deste período, mesmo que já tenha havido várias trocas em função desta compra.
No termo de garantia é anotado a data da compra, marca, modelo e número de série do aparelho vendido juntamente com o nome e endereço do cliente que o comprou. A cada troca de aparelho, relativo a primeira compra, é verificado se ainda está no prazo de garantia, e é registrado o cliente que realizou a troca. Qualquer cliente pode realizar uma troca, mesmo que não tenha sido o comprador. Os aparelhos defeituosos são devolvidos para a fábrica e não mais retornam para a loja. A loja quer saber:
a) Relação de aparelhos disponíveis na loja
b) Relação de aparelhos que apresentaram defeitos contendo quem realizou a troca, a data e o defeito apresentado
c) Relação de clientes cujas compras nunca apresentaram defeito

191. Uma firma que utiliza equipamentos de informática necessita de um sistema que gerencie a sua rede de microcomputadores (ponto-a-ponto, não existe servidor da rede), controlando usuários, máquinas e impressoras. A rede é composta de servidores de impressão, estações e impressoras. O sistema também irá controlar a partir de qual estação o usuário está conectado a rede, e os seus arquivos enviados para impressão.
Para todos os micros deseja-se cadastrar: código do patrimônio, descrição, capacidade do disco rígido, quantidade de memória e, sendo uma estação a sua localização, sendo um servidor o tamanho máximo do buffer e a quantidade máxima de buffers de impressão e que ele suporta, e as impressoras ligadas a ele (no máximo 3), caso existam. Para impressoras deseja-se cadastrar: código do patrimônio, descrição, velocidade (CPS) e, consequentemente, o servidor a que está ligada. Nesta firma todas as impressoras estão ligadas a algum servidor, não sendo compartilhada por mais de um servidor.
Para controlar os usuários, o sistema só precisa do nome de guerra e senha de cada um. Como os usuários não possuem máquina fixa, a sua conexão à rede pode ocorrer a partir de qualquer estação. Tendo o usuário uma conexão ativa, o sistema não permitirá que ele se conecte a partir de outra estação. Há interesse em controlar apenas as conexões ativas (as conexões desfeitas são irrelevantes).
No caso de impressão, o sistema deverá saber qual o arquivo, de quem ele é, e em qual impressora será impresso (atenção: somente usuários com conexão ativa e que possuem condição de enviar arquivos para impressão). É o usuário que escolhe a impressora onde ele quer que o seu arquivo seja impresso. Nada impede que usuários diferentes enviem arquivos de mesmo nome para impressão, porém (nesta firma) para o mesmo usuário isso não e possível, mesmo em impressoras diferentes. Neste caso o sistema permite alterar o número de cópias a serem impressas. Só deve ser mantido registro dos arquivos que ainda estão na fila de impressão.
O sistema deverá listar, para cada impressora, os arquivos que estão aguardando impressão, com o respectivo usuário que a enviou, mesmo que o usuário não esteja mais ativo na rede. Sempre que solicitado o sistema exibirá, para cada estação, o seu código e, caso exista, o nome do usuário conectado, a data e hora início desta conexão e, se houver, nome e quantidade de cópias dos arquivos que ele enviou e que ainda estão aguardando impressão.

192. Um Vídeo Clube deseja controlar o empréstimo de fitas a seus sócios. Atualmente o controle de empréstimo é feito utilizando dois conjuntos de fichas: o das informações dos

sócios e o das informações das cópias, no qual se controla também os empréstimos. Nas fichas dos sócios constam as seguintes informações: número de inscrição, nome, endereço e telefone. Nas fichas das cópias estão registrados: código de identificação da fita, título do filme, duração, ano, gênero, nome do diretor e de 2 (dois) dos artistas principais, data de aquisição e o estado da fita (bom ou ruim), além de uma lista de empréstimos com número de inscrição do sócio, data do empréstimo, data da devolução e valor pago. Estes dois últimos são preenchidos quando a cópia é devolvida.

Nesta locadora existem vários filmes com várias cópias, e cada cópia recebe uma etiqueta com um código, por exemplo: 5.315 (como se fosse o número do CONCINE, único para cada cópia original).

Um empréstimo é valido por 72 horas no máximo e possui preço fixo. A locadora pretende colocar terminais de consulta em sua loja, possibilitando seus sócios a escolha de filmes a partir do gênero, diretor ou dos artistas prediletos. Não é aceita reserva e todos os pagamentos são feitos no momento da devolução. Se o sócio atrasar a devolução deverá pagar multa.

A gerência deseja que o sistema forneça um relatório de todos os filmes que possui, informando a quantidade de cópias por filme, além de relatórios de filmes por gênero, diretor e artista. Ela deseja também uma lista das cópias em mal estado e outra de sócios inadimplentes. Existe interesse em manter um controle sobre diretores e artistas com nome, país de origem e data de nascimento de cada um.

193. O SUS está interessado em controlar os pacientes internados, e seus atendimentos, nos seus hospitais. Quando uma pessoa credenciada junto ao SUS passa mal, ela se dirige a um dos hospitais e se consulta com algum médico. Dependendo da gravidade o(s) médico(s) pode(m) decidir pela internação. Os pacientes, pessoas credenciadas que foram internadas, podem receber atendimento de vários médicos e enfermeiras durante o período de internação. Não há interesse em controlar as pessoas que não foram internadas, nem as consultas antes da internação.

Cada empregado do SUS (médico ou enfermeira) só pode estar vinculado a no máximo 3 hospitais. Não se admite um empregado com mais de um vínculo no mesmo hospital. Não há interesse em controlar as datas em que ocorreram os atendimentos. Há necessidade de se conseguir as seguintes informações:

a) Relação dos pacientes (nome, código do seguro social, idade) internados num hospital juntamente com os nomes e números dos médicos responsáveis por cada internação, e o período de internação.

b) Relação dos médicos e enfermeiras (nome, matricula) que trabalham determinado hospital.

c) Relação dos médicos (nome, matrícula, especialidade) e enfermeiras (nome, matrícula, cargo) que deram atendimento a um paciente durante uma internação.

d) Relação dos hospitais (nome, código e endereço) que um médico ou enfermeira mantém vínculo.

194. A administradora de Imóveis "IMOR Tal" é uma empresa que cuida, principalmente, da compra e venda de imóveis residenciais e comerciais no Grande Rio, dentre outras atividades. O atendimento atualmente e demorado e, muitas vezes, incompleto devido a demora no manuseio de muitas fichas, acarretando a perda de muitas oportunidades de negócio. Todos os imóveis são comprados pela imobiliária para, então, serem colocados a venda. A direção da empresa definiu como prioridade automatizar o processo de comercialização (compra e venda) dos imóveis, envolvendo seus proprietários (novos e antigos). A imobiliária considera "proprietário" toda pessoa que participou de um processo de comercialização (compra ou venda) no papel de dono (antigo ou novo). Entre outras

informações, o sistema deverá ser capaz de controlar os imóveis comprados, vendidos e os de seu “interesse”(não foram comercializados), e emitir:

a) Relação de todos os imóveis disponíveis para venda, contendo para cada um: Endereço, Bairro, Área (m2), descrição, Proprietário antigo (o atual é a administradora) e o Preço Mínimo para venda

b) Relação de todos os imóveis vendidos, por bairro, contendo para cada um: Bairro, Proprietário antigo, Proprietário novo, Preço de venda (ao proprietário novo) e o Preço de compra (pela imobiliária)

c) Relação dos proprietários que compraram mais de um imóvel na imobiliária (nome, CPF, endereço, telefone)

d) Relação dos proprietários que venderam mais de um imóvel para a imobiliária (nome, telefone)

195. Uma fábrica de roupas exclusivas (cada modelo, único, e projetado por estilistas famosos) deseja um sistema para controlar sua produção. A fábrica conta atualmente com 1230 funcionários sendo que a maior parte dos mesmos são costureiras trabalhando na atividade fim. A fabrica possui aproximadamente 600 máquinas de costura de diversos tipos (overlock, zig-zag, costura reta, etc.)de diversos fabricantes. Para ingressar como costureira, a funcionária é avaliada para determinar em que tipo de máquina ela possui habilitação. Cada máquina pode realizar um ou mais tipos de costura.

Cada peça de roupa é produzida integralmente por uma costureira em uma máquina, sendo que neste período nem a costureira, nem a máquina podem ser alocados para outra coisa.

A remuneração das costureiras e mensal baseado em uma alíquota fixa (15%) sobre o preço de venda de cada peça. Nenhuma costureira pode receber menos que um determinado valor mínimo que é negociado no momento da contratação de cada uma. As costureiras são divididas em supervisões, cada uma possuindo uma supervisora que é a responsável pela qualidade do que é produzido, e pela monitoração das máquinas que estão em conserto. A maquina só vai para conserto após o término da produção da peça. A supervisora da costureira que estava produzindo nesta máquina se torna a responsável pela monitoração de seu conserto.

A fábrica necessita das seguintes informações:

a) Relatório de peças produzidas por uma costureira num determinado período, no seguinte formato: modelo da peça, descrição do modelo, data e hora de início e término da fabricarão, código da máquina de costura, localização da máquina e o fabricante.

b) Relatório das costureiras sem produção no período (matrícula da costureira, nome, Valor Mínimo Negociado).

c) Relação das máquinas disponíveis, informando para cada uma o seu fabricante e o(s) tipo(s) de costura que possui.

d) Quais costureiras estão disponíveis e habilitadas a trabalhar em um tipo de máquina no momento?

e) Relação das máquinas que estiveram mais de 10 vezes em conserto, contendo: código da máquina e para cada conserto, matrícula e nome da supervisora responsável, data início e término do conserto.

Colaboração de Carlos Maurício