

ALGORITMOS E ESTRUTURA DE DADOS

2ª Lista de exercícios - 1 Sem/2021 – CCO – período – Piedade / Boa vista

1ª parte – Exercícios simples

01. Fazer um algoritmo que calcule e imprima o salário reajustado de um funcionário de acordo com as seguintes regras:

- Salário de até R\$ 500,00 reajuste de 50%;
- Salário maiores que R\$ 200,00 reajuste de 30%;

02. Uma empresa concederá um aumento de salário aos seus funcionários, variável de acordo com o cargo, conforme a tabla abaixo. Faça um algoritmo que leia o salário e o cargo de um funcionário e calcule o novo salário. Se o cargo do funcionário não estiver na tabela, ele deverá receber 40% de aumento. Mostre o salário antigo, o novo salário e a diferença.

Código	Cargo	Percentual
101	Gerente	10%
102	Engenheiro	20%
103	Técnico	30%

03. Desenvolver um algoritmo para calcular e imprimir o preço final de um carro. O valor do preço inicial de fábrica é fornecido por meio de entrada. O carro pode ter as seguintes opções:

- (a) Ar condicionado: R\$ 2.450,00
- (b) Pintura Metálica: R\$ 900,00
- (c) Vidro Elétrico: R\$ 1.800,00
- (d) Direção Hidráulica: R\$ 2200,00

04. Construa um algoritmo que calcule e escreva o custo e a quantidade de latas de tinta necessárias, para pintar tanques cilíndricos de combustível. É fornecida as seguintes informações: o preço da lata de 5 litros, o raio e a altura dos tanques cilíndricos. Cada litro de tinta pinta 3 metros quadrados.

05. Dada a atual crise hídrica do país, as pessoas começaram a construir reservatórios para armazenar água em suas propriedades. Faça um programa em linguagem C que auxilie os utilizadores do reservatório a controlarem seu consumo. Obtenha do teclado as dimensões de um reservatório (altura, largura e comprimento, em centímetros) e o consumo médio diário dos utilizadores do reservatório (em litros/dia). Assuma que o reservatório esteja cheio, tenha formato cúbico e informe:

(a) A capacidade total do reservatório, em litros;

(b) A autonomia do reservatório, em dias;

(c) A classificação do consumo, de acordo com a quantidade de dias de autonomia: Consumo elevado, se a autonomia for menor que 2 dias; Consumo moderado, se a autonomia estiver entre 2 e 7 dias; Consumo reduzido, se a autonomia maior que 7 dias.

Obs.: Considere que cada litro equivale a 1000 cm^3 ou 1 dm^3 .

2ª parte – Exercícios medianos

01. Fazer um programa para simular uma agenda de telefones. Para cada pessoa devem-se ter os seguintes dados:

- Nome
- E-mail
- Endereço (contendo campos para Rua, número, complemento, bairro, cep, cidade, estado, país).
- Telefone (contendo campo para DDD e número)
- Data de aniversário (contendo campo para dia, mês, ano).
- Observações: Uma linha (string) para alguma observação especial.

(a) Definir a estrutura acima.

(b) Declarar a variável agenda (vetor) com capacidade de agendar até 100 nomes.

(c) Definir um bloco de instruções busca por primeiro nome: Imprime os dados da pessoa com esse nome (se tiver mais de uma pessoa, imprime para todas).

(d) Definir um bloco de instruções busca por mês de aniversário: Imprime os dados de todas as pessoas que fazem aniversário nesse mês.

(e) Definir um bloco de instruções busca por dia e mês de aniversário: Imprime os dados de todas as pessoas que fazem aniversário nesse dia e mês.

(f) Definir um bloco de instruções insere pessoa: Insere por ordem alfabética de nome.

(g) Definir um bloco de instruções retira pessoa: Retira todos os dados dessa pessoa e desloca todos os elementos seguintes do vetor para a posição anterior.

(h) Definir um bloco de instruções imprime agenda com as opções:

- Imprime nome, telefone e e-mail.
- Imprime todos os dados.

(i) O programa deve ter um menu principal oferecendo as opções acima.

02. Faça um programa que gerencie o estoque de um mercado e:

- crie e leia um vetor de 5 produtos, com os dados: código (inteiro), nome (máximo 15 letras), preço e quantidade.
 - Leia um pedido, composto por um código de produto e a quantidade. Localize este código no vetor e, se houver quantidade suficiente para atender ao pedido integralmente, atualize o estoque e informe o usuário. Repita este processo até ler um código igual a zero.
- Se por algum motivo não for possível atender ao pedido, mostre uma mensagem informando qual erro ocorreu.

03. Elabore um algoritmo que preencha duas matrizes 8×8 com valores aleatórios 0 e 1 e em seguida, determine se as matrizes são iguais (possuem os mesmos valores em cada posição).
04. Faça um programa que leia um valor n e crie dinamicamente um vetor de n elementos e passe esse vetor para uma função que vai ler os elementos desse vetor. Depois, no programa principal, o vetor preenchido deve ser impresso. Além disso, antes de finalizar o programa, deve-se liberar a área de memória alocada.
05. Faça uma função que receba um valor n e crie dinamicamente um vetor de n elementos e retorne um ponteiro. Crie uma função que receba um ponteiro para um vetor e um valor n e imprima os n elementos desse vetor. Construa também uma função que receba um ponteiro para um vetor e libere esta área de memória. Ao final, crie uma função principal que leia um valor n e chame a função criada acima. Depois, a função principal deve ler os n elementos desse vetor. Então, a função principal deve chamar a função de impressão dos n elementos do vetor criado e, finalmente, liberar a memória alocada através da função criada para liberação.
06. A linguagem C permite alocar (reservar) dinamicamente (em tempo de execução) blocos de memórias utilizando ponteiros. A esse processo dá-se o nome de alocação dinâmica, que faz uso das funções malloc, calloc, realloc e free, disponíveis na biblioteca stdlib.h. Para liberar um bloco de memória previamente alocado, por meio de um único parâmetro de entrada, faz-se uso de qual função?
- a) Malloc.
 - b) Calloc.
 - c) Realloc.
 - d) Free.
07. Criar uma função que recebe dois números complexos como parâmetro e retorna um ponteiro para uma área dinamicamente criada, contendo a soma destes números. A função retorna NULL caso não tenha sido possível alocar a área para a soma.