

Lista de Exercícios 2 – Programação Estruturada

I - Sobre Estruturas

1. Crie uma estrutura chamada "pessoa" com os campos "nome", "idade" e "altura". Em seguida, crie uma variável do tipo "pessoa" e atribua valores aos seus campos.
2. Crie uma união chamada "tipo_dado" que pode armazenar um número inteiro, um número de ponto flutuante ou uma string. Em seguida, crie uma variável do tipo "tipo_dado" e atribua valores a ela.
3. Crie um enum chamado "dia_semana" com os valores "SEGUNDA", "TERCA", "QUARTA", "QUINTA", "SEXTA", "SABADO" e "DOMINGO". Em seguida, crie uma variável do tipo "dia_semana" e atribua o valor "TERCA" a ela.
4. Crie um enum chamado "tipo_animais" com os valores "CACHORRO", "GATO" e "PASSARO". Em seguida, crie uma estrutura chamada "animal" com os campos "tipo" (do tipo "tipo_animais") e "nome" (do tipo string). Em seguida, crie uma variável do tipo "animal" e atribua valores a ela.
5. Crie uma estrutura chamada "aluno" com os campos "nome", "idade" e "notas" (um array com as notas do aluno). Em seguida, crie uma função chamada "calcula_media" que recebe uma variável do tipo "aluno" e retorna a média das notas.
6. Crie uma união chamada "tipo_pessoa" que pode armazenar uma variável do tipo "pessoa" (da questão 1) ou uma variável do tipo "animal" (da questão 6). Em seguida, crie uma variável do tipo "tipo_pessoa" e atribua valores a ela.
7. Crie um enum chamado "tipo_figura" com os valores "CIRCULO", "QUADRADO" e "RETANGULO". Em seguida, crie uma estrutura chamada "figura" com os campos "tipo" (do tipo "tipo_figura") e "dados" (do tipo "void *", que pode ser um ponteiro para uma estrutura "circulo", "quadrado" ou "retangulo"). Em seguida, crie uma variável do tipo "figura" e atribua valores a ela.
8. Crie uma estrutura chamada "tempo" com os campos "hora", "minuto" e "segundo". Em seguida, crie uma função chamada "converte_segundos" que recebe uma variável do tipo "tempo" e retorna a quantidade de segundos correspondente a esse tempo.
9. Considere uma loja de eletrônicos que vende três tipos de produtos: smartphones, notebooks e televisores. Cada produto possui as seguintes informações: nome, marca, modelo e preço. Além disso, cada produto possui informações específicas, descritas a seguir:
 - Smartphone: sistema operacional (Android ou iOS) e capacidade de armazenamento (em GB).
 - Notebook: tamanho da tela (em polegadas) e quantidade de memória RAM (em GB).

- Televisor: tamanho da tela (em polegadas) e resolução (HD, Full HD, 4K).
 - a) Crie uma estrutura de dados chamada "Produto" que armazene todas as informações descritas acima. Utilize as estruturas "enum" e "union" quando necessário.
 - b) Crie uma função que receba como parâmetro um produto e imprima na tela todas as suas informações, incluindo as informações específicas de cada tipo de produto.
 - c) Crie um programa que permita ao usuário cadastrar produtos na loja, imprimir na tela os produtos cadastrados e buscar um produto pelo seu nome.

10. Considere o seguinte problema: uma empresa possui diversos funcionários que podem ser de diferentes níveis (júnior, pleno ou sênior) e ter diferentes tipos de contrato (CLT ou PJ). Cada funcionário é identificado por um código e possui um salário base.

Defina uma enumeração chamada "Nivel" com os valores JUNIOR, PLENO e SENIOR. Defina uma estrutura chamada "CLT" contendo os campos: salário bruto, vale-refeição e vale-transporte.

Defina uma estrutura chamada "PJ" contendo os campos: valor por hora trabalhada, número de horas trabalhadas e valor do benefício.

Defina uma union chamada "Contrato" que pode armazenar um funcionário do tipo "CLT" ou "PJ".

Defina uma estrutura chamada "Funcionario" contendo os campos: código, nome, nível, contrato e salário líquido. O campo "salário líquido" deve ser calculado a partir do salário base, descontando os impostos (20%) e adicionando eventuais benefícios.

Escreva uma função em C que receba como entrada um array de "Funcionarios" e o número total de funcionários, e retorne o salário médio líquido da empresa.

Exemplo do array de struct funcionarios:

```
struct Funcionario {
    int codigo;
    char nome[50];
    enum Nivel nivel;
    union Contrato contrato;
    float salario_base;
    float salario_liquido;
};

struct Funcionario funcionarios[] = {
    {1, "João", PLENO, .contrato.clt = {3500, 500, 300}, .salario_base = 5000},
    {2, "Maria", JUNIOR, .contrato.pj = {50, 160, 100}, .salario_base = 2500},
    {3, "Pedro", SENIOR, .contrato.clt = {4500, 600, 400}, .salario_base = 6000},
    {4, "Ana", JUNIOR, .contrato.pj = {55, 120, 80}, .salario_base = 2000},
    {5, "Lucas", SENIOR, .contrato.clt = {5500, 700, 500}, .salario_base = 8000}
};
```

II - Sobre alocação dinâmica de memória

1. Escreva uma função que receba como entrada um número inteiro N e retorne um array de N inteiros alocado dinamicamente. O array deve ser preenchido com os números de 1 a N.
2. Escreva uma função que receba como entrada dois arrays de inteiros de tamanho N, a e b, e retorne um novo array de inteiros alocado dinamicamente que seja a soma elemento a elemento de a e b.
3. Escreva uma função que receba como entrada um array de N inteiros alocado dinamicamente e retorne o índice do maior elemento desse array.
4. Escreva uma função que receba como entrada um array de N strings alocado dinamicamente e retorne uma nova string que seja a concatenação de todas as strings do array, na ordem em que aparecem.
5. Escreva uma função que receba como entrada uma string s e um caractere c, e retorne um array de inteiros alocado dinamicamente contendo as posições de todas as ocorrências de c em s.
6. Implemente um sistema de cadastro de funcionários de uma empresa. Cada funcionário possui as seguintes informações: Nome, Sobrenome, Idade e Salário. Defina a estrutura de dados para um funcionário. Permita que o usuário cadastre vários funcionários em um array de struct (alocar memória). O usuário deve ter a opção de visualizar todos os funcionários cadastrados
7. Escreva um programa que leia um número inteiro n do usuário e aloque dinamicamente um array de n inteiros. Em seguida, o programa deve preencher o array com valores lidos do usuário e calcular a média dos valores armazenados no array. Por fim, o programa deve imprimir a média na tela e liberar a memória alocada.
8. Escreva um programa que permita que o usuário insira uma string e depois a exiba invertida na tela. O programa deve alocar memória dinamicamente para armazenar a string e, em seguida, inverter a string usando um loop e alocando memória dinamicamente para armazenar a string invertida. O programa deve imprimir a string invertida na tela antes de liberar a memória alocada dinamicamente.

III – Sobre Manipulação de arquivos

1. Escreva um programa que crie um arquivo de texto chamado "ex1.txt" e escreva nele a frase: "É muito bom aprender linguagem de programação C !".
2. Desenvolva um programa que leia um arquivo de texto chamado "ex1.txt" e conte quantas vezes cada vogal aparece nele.
3. Implemente um programa que leia um arquivo de texto chamado "ex1.txt" e crie um novo arquivo chamado "exe3_copia.txt" com o mesmo conteúdo do arquivo original.
4. Escreva um programa que leia um arquivo de texto chamado "ex4.txt" e mostre na tela as linhas que começam com a letra "a".
5. Faça um programa que leia um arquivo de texto chamado "ex5.txt" e substitua todas as ocorrências da palavra "gato" pela palavra "cachorro" no arquivo original.
6. Crie um programa que leia um arquivo binário chamado "ex6.bin" contendo um array de inteiros e mostre na tela o valor médio dos números contidos no array.
7. Escreva um programa que leia um arquivo de texto chamado "ex7.txt" contendo nomes de pessoas e suas idades, separados por vírgula, e crie um novo arquivo chamado "ex7_idades.txt" contendo apenas as idades das pessoas.
8. Faça um programa em que recebe um arquivo "ex8.txt" com os nomes e as duas notas de alunos separados por vírgula e implemente uma função que permita ao usuário buscar pelo nome e possa modificar esses dados.
9. Faça um programa para abrir um arquivo de texto e que usa a função `fgetc()` e `fputc()` e substituir os caracteres: 'a' por 4, 'e' por 3, 'i' por 1, 'o' por 0 e 'u' por 5. Use a função `fseek()` para alterar esses caracteres no arquivo original.
10. Faça um programa que receba de um arquivo.txt com dados de alunos no seguinte formato:
nome1, nota 1, nota 2
nome2, nota 1, nota 2
O programa deve elaborar um relatório em outro arquivo com a lista de nomes, média das duas notas de cada aluno e situação: aprovado, na final ou reprovado. No final do arquivo informe porcentagem dos alunos aprovados e média da turma

IV – Sobre tempo de vida e escopo de variáveis e funções

1. O que é tempo de vida de uma variável e como é determinado em C?
2. O que é escopo de uma variável e como é determinado em C?
3. Qual a diferença entre uma variável fixa (static) e uma variável automática em C?
4. Escreva um programa em C que calcule a média de uma lista de números lidos do usuário. Utilize uma variável static para armazenar o número total de números lidos.
5. Escreva um programa em C que conte o número de vezes que uma determinada letra aparece em uma string lida do usuário. Utilize uma variável static para armazenar o número total de letras contadas.
6. Escreva um programa em C que imprima na tela a tabuada de multiplicação de 1 a 10. Utilize uma variável static para armazenar o número de vezes que a tabuada foi impressa.
7. Escreva uma função em C que receba um número inteiro positivo como parâmetro e retorne o seu fatorial. Utilize uma variável static para armazenar o resultado.
8. Escreva uma função em C que receba uma string como parâmetro e retorne o número de vogais que ela contém. Utilize uma variável static para armazenar o número total de vogais contadas. Além disso, utilize uma função static para determinar se um caractere é uma vogal ou não.