



Técnicas de Programação
[AN32F-IF62C]

PROFESSOR

Diogo Cezar Teixeira Batista

LISTA DE EXERCÍCIOS - 03

ALOCÇÃO DINÂMICA, MANIPULAÇÃO DE ARQUIVOS E RECURSIVIDADE

Cornélio Procópio

2017

1 INSTRUÇÕES

- Essa lista de exercícios deverá ser entregue em um arquivo único compactado no formato .zip ou .rar;
- Cada exercício deverá estar separado em um arquivo com o seguinte padrão de nomenclatura: L03_EXON.c onde N é o número da questão;
- O trabalho é individual;
- Todo arquivo fonte deve ter como comentário o nome do aluno, da turma e o que o exercícios faz;
- Cada aluno poderá ser convocado para explicar um dos exercícios resolvidos da lista para o professor, ou para toda a turma;
- Trabalho iguais serão anulados;
- Não se esqueça de organizar, e indentar o seu código;
- Utilize nomes significativos para as suas variáveis;
- Escreva as instruções para o usuário do programa que está criando;

2 EXERCÍCIOS

1. Crie um programa com o seguinte menu de opções:

```
<1> Cadastrar Cliente  
<2> Consultar Nome  
<3> Listar Todos  
<4> Salvar Como Arquivo Texto  
<5> Recuperar Arquivo Texto  
<6> Salvar Como Arquivo Binário  
<7> Recuperar Arquivo Binário  
<8> Sair
```

Cada uma destas opções, com exceção de sair, chama uma função que executará o que ela indica. Este programa irá cadastrar os seguintes dados do cliente: Nome, Idade, Rua, Bairro, Cidade, CEP e Telefone, todos estes campos devem fazer parte de uma struct que será utilizada para resolver o problema. O número máximo de

clientes que podem ser cadastrados no sistema será perguntado no início da execução do programa. Caso este limite seja alcançado, no momento de um novo cadastro, deverá ser impresso na tela uma mensagem de erro. Deve-se utilizar alocação dinâmica para armazenar os elementos em um vetor de registros. Na opção <2> deverá ser consultado um nome de cliente, caso seja localizado, sua ficha e impressa na tela. Na opção <3> é impresso o nome, idade, e telefone de todos os clientes cadastrados. Deve-se utilizar manipulação de arquivos para realizar as operações respectivas dos itens <4>, <5>, <6> e <7>. Deve-se utilizar os nomes de arquivos: cliente.txt e cliente.bin para armazenar e recuperar os dados. Você poderá escolher a forma como irá percorrer os arquivos para realizar as operações de leitura e escrita.

2. Faça um programa que calcule a média de idades de uma turma. Primeiro deve-se perguntar quantas pessoas tem na turma, e a partir desta resposta alocar dinamicamente na memória um vetor de inteiros que suporte este total. Caso a alocação de errado mostre uma mensagem de erro e saia do programa. Caso contrário, devem-se ler as idades, listar todas, e calcular a média geral. Cada uma destas ações deve ser feita em uma função separada. Ao final antes de sair do programa libere a memória usada pelo vetor.
3. Faça um programa que calcule recursivamente a soma dos números no intervalo de 1 a N, ou seja $(1 + 2 + 3 + 4 \dots + N)$. O Valor N deve ser inteiro e digitado pelo usuário.
4. Implemente um programa que dados dois números inteiros X e N, calcule recursivamente o valor de X elevado a N.
5. Faça um programa que contenha uma função recursiva para calcular e retornar o resultado da seguinte série:

$$\frac{1}{(n)} + \frac{2}{(n-1)} + \frac{3}{(n-2)} + \frac{4}{(n-3)} + \dots$$

N é um valor inteiro maior ou igual a 1 digitado pelo usuário. A série deve ser calculada até que o denominador seja igual a 1. O valor de N deverá ser lido com o usuário e armazenado numa variável global. Dica, o parâmetro da função pode ser o valor que está sendo diminuído do denominador.