Linguagem de Programação

Prof. Isaac Benchimol

Lista - Arquivos

1. Faça um programa que gere uma versão criptografada de um arquivo texto trocando cada caractere de código ASCII j pelo caractere de código ASCII j+k, onde k é um parâmetro especificado pelo usuário. Fique atento para não gerar códigos ASCII fora da faixa permitida (0 a 255).
2. Escreva um programa em C, que lê a partir de um arquivo de entrada, duas matrizes descritas da seguinte forma: Os dois primeiros valores, para cada matriz, identificam respectivamente o número de linhas e colunas, e os demais elementos são os valores a serem inseridos na matriz, sendo que a mesma deve ser preenchida no sentido das linhas. A partir da leitura das matrizes, o programa deverá ler uma operação matemática básica (+ - ou \*) e verificar se a operação poderá ser realizada, de acordo com as dimensões das matrizes lidas. Se a operação puder ser realizada, todos os elementos correspondentes nas matrizes devem ser operados e as matrizes originais e a matriz resultado devem ser impressas na tela.
3. Faça um programa para ler dois arquivos tipo texto, “arq1.txt” e “arq2.txt”, e mostre as linhas comuns em ambos os arquivos.

Exemplo:

**arq1.txt arq2.txt**

eu estudo no Ifam eu estudo no Ifam

ola, como vai? como vai voce?

Ate logo Ate logo

Ecat ecat

A saída do programa deverá ser:

**Linha 1: eu estudo no Ifam**

**Linha 3: Ate logo**

1. Faça um programa para ler o arquivo-texto **alunos.txt** que contém informações sobre os alunos de uma turma conforme a classe

class Aluno {

char nome[50];

float n1;

float n2;

public:

float media();

void print();

};

Para cada aluno, calcule a média e a situação (aprovado ou reprovado). Considere aprovado o aluno que obtiver média igual ou superior a 6,0. A função print() deverá mostrar em uma única linha para cada aluno, o seu nome, as notas, a média e a situação conforme o formato abaixo:

[Nome] [n1] [n2] [Média] [Situação]

Exemplo:

**[Carlos Silva] [5.0][8.0][6.5][APROVADO]**

1. Considere a classe Conta de uma instituição bancária

class Conta {

public:

int numconta;

char nome[30];

double saldo;

};

1. Faça um programa para criar o arquivo binário contas **contas.dat** com 50 objetos da classe Conta com os seguintes valores iniciais: numconta = 1 a 50, nome = “sem nome” e saldo = 0;
2. Faça um programa para manipular o arquivo **contas.dat**. Crie um menu para gerenciar as contas com as opções: inserir, consultar, listar e remover.
3. Na opção inserir, os dados da conta deverão ser escritos no arquivo contas.dat na posição correspondente ao número da conta. As opções listar e remover devem operar sobre as contas inseridas e efetuar as respectivas funções.
4. As contas são inseridas em qualquer ordem de número, por exemplo:

Opção **inserir**:

Entre número da conta (1 a 50)

? 27

Entre nome para conta 27

? Fabio Souza

Entre saldo para conta 27

? 1500

Opção **listar** (considerando as contas 2, 8 e 27 inseridas):

Conta 2: Carlos Andre 4500.23

Conta 8: Ana Rosa 6500.45

Conta 27: Fabio Souza 1500

Opção **consultar** (considerando contas 2, 8 e 27 inseridas):

Entre número da conta (1 a 50)

? 27

Conta 27: Fabio Souza 1500

Opção **consultar** (considerando contas 2, 8 e 27 inseridas):

Entre número da conta (1 a 50)

? 20

Conta 20: nao cadastrada

Opção **remover** (considerando contas 2, 8 e 27 inseridas):

Entre número de conta (1 a 50)

? 27

Conta 27: Fabio Souza 1500

Remover (S/N): S

Conta 27 removida (nome = “sem nome” e salario = 0)

Opção **remover** (considerando contas 2, 8 e 27 inseridas):

Entre número de conta (1 a 50)

? 20

Conta 20: nao cadastrada

O arquivo **contas.dat** deve manter os dados válidos para ser utilizado pelo programa de manipulação sempre que necessário.