MC 102 - Algoritmos e Programação de Computadores Primeiro Semestre de 2012

Lista de Exercícios

28 de maior de 2012

1. A cifra de César é uma forma de criptografia antiga, onde cada letra de uma determinada mensagem é trocada pela letra que a sucede em 3 posições. Por exemplo, a palavra **ajuda** seria transformada em **dmxgd**. Escreva uma função que leia um arquivo texto e escreva a cifra de César correspondente em um outro arquivo texto.

2. Faça uma função que leia um arquivo texto contendo uma cifra de César (veja o exercício anterior). A função deve decifrar a mensagem e escrever o resultado em um arquivo texto.

3. Adapte o programa feito no exercício anterior para que ele receba por parâmetro um número inteiro. Este inteiro deve estar no intervalo entre 0 e 26, representando a quantidade posições que devem ser puladas para criptografar uma letra.

4. Implemente um programa que receba por parâmetro o nome de um arquivo que será lido. O conteúdo do arquivo deve ser impresso na tela e o usuário poderá escolher ser quer apagar este arquivo ou não.

5. Neste exercício use um arquivo binário para armazenar as informações nome, salário e número de identificação. Estas informações são usadas para controlar os empregados de uma empresa. Para cada empregado, use a estrutura a seguir:

```
struct employee {
  long id;
  char name[50];
  double salary;
};
```

   Implemente as seguintes funções:

   (a) `int add (fname, empId, stringName, salary);`

   onde fname é uma string contendo o nome do arquivo, empId é um inteiro, stringname representa o nome do empregado, e salary é uma variável do tipo double, contendo o salário. Esta função acrescenta um empregado no

final do arquivo binário (empId é uma chave que identifica univocamente o empregado, portanto não podem ter dois empregado com mesmo valor de empId)

(b) `void moreDollars (fname, empId, incr);`

com 3 parâmetros: uma string fname, um inteiro empId, e uma variável incr, representando o incremento salarial de um determinado empregado.

(c) `void show (const char *fname);`

que mostra todas as informações contidas no arquivo.

6. Faça um programa que leia um arquivo binário contendo números inteiros separados por #, ordene os inteiros e escreva o resultado da ordenção em um arquivo binário novo.

7. Altere o programa do exercício anterior para que sejam eliminadas repetições de números.

8. Faça um programa que leia um arquivo binário e imprima na tela o valor hexadecimal equivalente a cada byte que foi lido. Por exemplo, o byte 0101 1101 representa o valor 5D em hexadecimal.

9. Altere o programa do exercício anterior, para que seja passado por parâmetro um número inteiro k. O conteúdo do arquivo deverá ser impresso na tela usando base k. Ou seja, o valor 0101 1101 na base 10 resultaria em 93.

10. Faça um programa que leia um arquivo binário contendo números inteiros de 4 bytes. Verifique se todos os números pertencem ao intervalo de -32768 a 32767, caso negativo apresente uma mensagem de erro, caso positivo, escreva os mesmos números em um arquivo binário, mas usando o tipo short ao invés de int. Verifique que o arquivo foi comprimido em 50%.