

Introdução à Programação para Engenharia da Computação

Professor: Adriano Sarmento

Quarta Lista de Exercícios

Data de Entrega: 22/10/2009

PONTEIROS

Questão 1:

Escreva um programa que lê até n (dado pelo usuário) strings e classifica-as em ordem alfabética, utilizando a função "strcmp" <string.h>. As strings são terminadas por um '\0' caractere. A classificação é feita mantendo um vetor de ponteiros para as strings que simplesmente troca os ponteiros até que a ordem esteja correta. Isto economiza ter de copiar as strings, o que melhora a eficiência.

Questão 2:

Um ponteiro pode ser usado para dizer a uma função onde ela deve depositar o resultado de seus cálculos. Escreva uma função “relogio” que converta minutos em horas-e-minutos. A função recebe um inteiro mnts e os endereços de duas variáveis inteiras, digamos h e m, e atribui valores a essas variáveis de modo que m seja menor que 60 e que $60*h + m$ seja igual a mnts. Escreva também uma função main que use a função relogio.

Questão 3:

Fazer uma função FATORES que:

- Recebe 3 parâmetros: um vetor de inteiros, um número inteiro n passado por valor e outro número x passado como ponteiro.
- Retorna um número inteiro.
- Decompõe o número n em fatores primos e armazena-os nas posições do vetor. O conteúdo de x deve receber o número de fatores primos encontrados. Caso o número de fatores encontrados seja maior que 5, a função deve retornar 1, do contrário deve retornar 0, você não deve contar fatores repetidos.

Questão 4:

A Google está pesquisando um novo algoritmo de busca. Pra isso, fez um concurso para os alunos de computação e irá escolher o algoritmo mais eficiente que for submetido nesse concurso.

Esse algoritmo terá que procurar dentro de arquivos a palavra chave que for pedida pelo usuário. Pronto pra começar?

Tarefa

É dado um texto. Você deverá procurar nesse texto a palavra chave dada, ou palavras que contenham essa palavra, e deve determinar quantas vezes e em quais linhas ela aparece.

Entrada

Seu programa deverá pedir uma palavra chave, e a seguir o numero de linhas do texto. Feito isso as linhas deverão ser recebidas. O tamanho máximo das linhas é de 100 caracteres.

Exemplo de Entrada:

Paralelepípedo 11
Disseram que na minha rua
Tem paralelepípedo feito
De paralelogramos.
Seis paralelogramos
Tem um paralelepípedo.
Mil paralelepípedos
Tem uma paralelepípedovia.
Uma paralelepípedovia
Tem mil paralelogramos.
Então uma paralelepípedovia
É uma paralelogramolândia?

Saída

A saída deve conter o número de vezes que a palavra aparece no texto e em quais linhas ela aparece.

Exemplo de Saída:

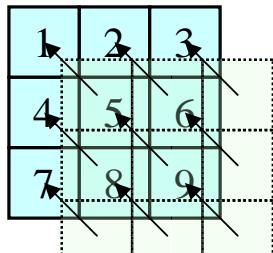
Palavra Chave paralelepípedo aparece no texto 6 vezes, na(s) linha(s) 2,5,6,7,8,10,11.
(esta saída corresponde ao exemplo de entrada acima)

Observações:

- Não fazer diferença entre maiúsculas e minúsculas.
- Imprimir “palavra chave inexistente” se não encontrar a palavra chave no texto.

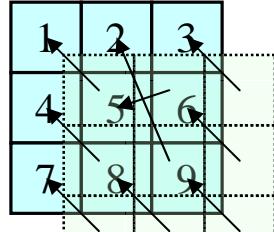
Questão 5:

Você fará uma matriz de ponteiros que aponta para uma outra matriz sendo esta de inteiros, seu programa deverá primeiro pedir a dimensão das matrizes, e a seguir receber cada elemento da matriz de inteiros, depois deverá imprimir um menu com as seguintes opções de operações:



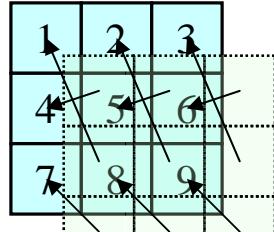
Exemplo de matriz de ponteiros apontando para uma de inteiros.

- **Swap:** pede duas posições (i,j), e faz com que os elementos correspondentes da matriz de ponteiros troquem de posição.



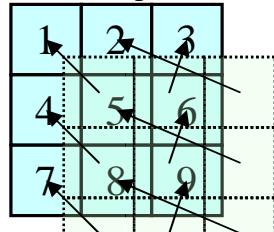
Resultado da Operação swap das posições (0, 1) e (1, 1).

- **Swap Row:** pede duas linhas i e faz com que os elementos correspondentes da matriz de ponteiros troquem de posição.



Resultado da operação Swap Row das linhas 0 e 1.

- **Swap Column:** pede duas colunas j e faz com que os elementos correspondentes da matriz de ponteiros troquem de posição.



Resultado da operação Swap Column das colunas 1 e 2.

- **Print:** imprime a matriz original de inteiros, e a matriz dos valores correspondentes a cada elemento da matriz de ponteiros.
- **Sair:** Sai do programa.

Obs1: Não deverá ser usada alocação estática para as matrizes, use as funções malloc ou calloc e não esqueça de liberar a memória alocada.

Obs2: O programa só deverá terminar se a opção sair do menu for escolhida.