III IME SOFTEST

23 de agosto de 1997

Maratona de Programação

Caderno de Questões

Problema 1:
Nome do fonte: **inverte.pas** ou **inverte.c**
Arquivo de entrada: **inverte.in**
Arquivo de saída: **inverte.out**

Faça um programa que receba uma seqüência de letras e brancos, terminada por '.', e inverta a ordem das palavras da seqüência. Na saída, seqüências de brancos deverão ser comprimidas para apenas um branco.

Exemplo: Dada a seqüência,

OLÁ ESTE É   UM EXEMPLO SEM   GRAÇA.

a saída deverá ser

GRAÇA SEM EXEMPLO UM É ESTE OLÁ

Problema 2
Nome do fonte: **encaixa.pas** ou **encaixa.c**
Arquivo de entrada: **encaixa.in**
Arquivo de saída: **encaixa.out**

Dizemos que uma palavra u **encaixa** em uma palavra v, se v for da forma aub onde a e b são palavras não vazias. Assim,

- bolo encaixa em embolotado
- caixa não encaixa em encaixa
- carro não encaixa em carroça
- menta encaixa em aumentar

Faça um programa que receba uma seqüência de letras e brancos, terminada por '.' e liste os pares de palavras encaixantes da seqüência.

Problema 3
Nome do fonte: **matrizes.pas** ou **matrizes.c**
Arquivo de entrada: **matrizes.in**
Arquivo de saída: **matrizes.out**

Faça um programa que leia um inteiro $n \geq 1$ e duas matrizes de números reais $A_{n\times n}$ e $B_{n\times n}$ e verifica se alguma(s) das possibilidades abaixo ocorre:

- A é uma cópia de B;
- A é uma versão espelhada de B; onde o espelho pode ser colocado em qualquer das quatro bordas de A;
- A é uma rotação das linhas de B;
  Ex.: $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ é uma rotação das linhas de $\begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \end{pmatrix}$.
- A é uma rotação das colunas de B.
  Ex.: $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ é uma rotação das colunas de $\begin{pmatrix} 3 & 1 & 2 \\ 6 & 4 & 5 \\ 9 & 7 & 8 \end{pmatrix}$.

Problema 4
Nome do fonte: **filmes.pas** ou **filmes.c**
Arquivos de entrada: **filmes.in** e **generos.in**
Arquivo de saída: **filmes.out**

São dadas no arquivo de entrada `filmes.in` informações sobre nomes (string de 30 caracteres), diretores (string de 30 caracteres) e gêneros (string de 15 caracteres) de filmes feitos para o cinema. Além disso, é dado um arquivo `generos.in` com uma lista de gêneros de filmes (comédia, aventura, terror, etc).

Seu programa deverá listar o nome dos diretores que fizeram pelo menos um filme de cada um dos gêneros existentes (listados no segundo arquivo).

Problema 5
Nome do fonte: **caminho.pas** ou **caminho.c**
Arquivos de entrada: **caminho.in**
Arquivo de saída: **caminho.out**

Escreva um programa que receba inteiros $m \geq 1$ e $n \geq 1$ e uma matriz $M_{m \times n}$ de inteiros positivos e encontre um caminho do canto superior direito até o canto inferior esquerdo da matriz, de forma a minimizar a soma dos valores absolutos das diferenças entre elementos consecutivos do caminho.

Um caminho é definido por movimentos verticais ou horizontais entre elementos adjacentes da matriz. Assim, para a matriz 5×5 abaixo, o melhor caminho está sublinhado e tem custo total 66.

$$\begin{pmatrix} \underline{1} & \underline{17} & \underline{6} & 46 & \underline{7} \\ \underline{4} & 58 & \underline{15} & \underline{6} & \underline{10} \\ \underline{2} & 8 & 61 & 18 & 29 \\ \underline{9} & 18 & 6 & 23 & 9 \\ \underline{11} & 6 & 13 & 41 & 12 \end{pmatrix}$$

Problema 6
Nome do fonte: **distr.pas** ou **distr.c**
Arquivos de entrada: **distr.in**
Arquivo de saída: **distr.out**

Em um sistema distribuído, m máquinas estão disponíveis para executar as n tarefas. As tarefas devem ser executadas simultaneamente. Cada máquina tem capacidade de memória de F MBytes, e cada tarefa i exige f_i MBytes para ser executada. Cada máquina pode executar várias tarefas ao mesmo tempo, desde que a restrição de capacidade seja respeitada. Além disso, as tarefas se comunicam bastante, e para cada par de tarefas (i, j) é dada a taxa esperada de comunicação entre as duas tarefas. Todos os dados são números inteiros e positivos.

Seu programa deve ler essas informações e determinar uma forma de alocar as tarefas aos computadores de forma que a capacidade de memória destes não seja excedida e, além disso, a taxa esperada de comunicação entre máquinas seja minimizada.