Descrição do Problema

IMPLEMENTE UTILIZANDO SHELLSORT OU QUICKSORT

A Secretaria Municipal de Saúde de Ouro Preto/MG precisa alocar médicos nas Unidades de Pronto Atendimento (UPA 24h) de modo a otimizar a oferta de atendimento para a população em um determinado dia da semana. Seu programa deve ajudar os gestores nessa tomada de decisão, listando as UPAs em ordem de prioridade.

É importante salientar que os gestores avaliam a quantidade de pacientes em espera nas unidades, considerando a classificação da gravidade definida durante a triagem. A classificação da gravidade se baseia no Protocolo de Manchester (adaptado) como mostra a Figura 1. Os gestores devem priorizar as UPAs com maior número de pacientes em espera para Emergência, Urgência e Sem Urgência, respectivamente. Caso duas unidades tenha o mesmo número de pacientes para Emergência, deve-se considerar a demanda subsequente e assim sucessivamente. Caso os quantitativos entre duas UPAs sejam exatamente iguais, os gestores optam por aquela com menor número de médicos de plantão. Se persistir o empate, as UPAs devem ser listadas em ordem alfabética.

Detalhes da implementação

O código-fonte deve ser modularizado corretamente em três arquivos: principal.c, ordenacao.h e ordenacao.c. O arquivo principal.c deve apenas invocar as funções e procedimentos definidos no arquivo ordenacao.h. A separação das operações em funções e procedimentos está a cargo do aluno, porém, não deve haver acúmulo de operações dentro de uma mesma função/procedimento. Além disto, as informações de cada UPA devem ser armazenadas em um tipo abstrato de dados criado especificamente para isso. O conjunto de informações de todas as UPAS deve ser armazenado em um vetor alocado dinamicamente (e posteriormente desalocado) para cada caso de teste. O aluno pode escolher qual algoritmo de ordenação será utilizado para abordar o problema, entretanto, o limite de tempo para solução de cada caso de teste é de apenas um segundo. Utilize suas habilidades de programação e de análise de algoritmos para desenvolver um algoritmo correto e rápido!

Entrada

A entrada é dada pelo número de UPAs a serem avaliadas, seguido pela lista dos nomes das UPAs (sem acentos ou caracteres especiais), com as quantidades de pacientes em espera para Emergência, Urgência e Sem Urgência, além do total de médicos plantonistas.

Entrada no Terminal

```
5
UPANorte 2 5 2 3
UPASul 2 4 0 1
UPALeste 3 7 2 1
UPAOeste 3 7 2 2
UPACentral 3 7 2 2
```

Saída

A saída deve ser a lista de UPAs em ordem de prioridade com seus respectivos quantitativos de pacientes em Emergência, Urgência e Sem Urgência e o total de médicos plantonistas.

Saída esperada no Terminal

```
UPALeste 3 7 2 1
UPACentral 3 7 2 2
UPAOeste 3 7 2 2
UPANorte 2 5 2 3
UPASul 2 4 0 1
```

Diretivas de Compilação

```
$ gcc -c ordenacao.c -Wall
$ gcc -c principal.c -Wall
$ gcc ordenacao.o principal.o -o exe
```