

Tarefa 6 - Sala de espera do Cecom

Prazo de entrega: 08/11/2020

[Voltar para lista de tarefas](#)

O Cecom, Centro de Saúde da Comunidade da UNICAMP, é um serviço oferecido a estudantes e funcionários da UNICAMP. Devido ao alto volume de usuários, um sistema de senhas e prioridades é usado para organizar os diferentes atendimentos oferecidos.

Há 9 especialidades diferentes de atendimento, oferecidas pelos diferentes profissionais no hospital. Cada especialidade possui um número identificador e uma quantidade de profissionais disponíveis no hospital.

ID	Atendimento	Número de profissionais
1	Clínico Geral	10
2	Radiologista	2
3	Enfermeiro (exame de sangue)	5
4	Otorrinolaringologista	3
5	Pneumologista	4
6	Dermatologista	7
7	Neurologista	2
8	Oncologista	1
9	Psiquiatra	4

Seu trabalho é implementar um sistema capaz de direcionar os pacientes a cada profissional, conforme ele fica disponível.

Entrada

A entrada é composta por uma lista de pacientes na ordem de chegada no hospital. Cada linha possui o nome de um indivíduo, a prioridade de atendimento e uma sequência de números que representa os atendimentos de que o paciente precisa. Veja o exemplo de uma linha:

```
"Ze Carioca" normal 1 5 3 1
```

Cada número da sequência de um paciente corresponde ao identificador de uma especialidade. Assim, ao chegar no hospital, Zé Carioca entrará numa fila de espera para profissionais do tipo 1, ao fim desse atendimento, ele entrará na fila de espera para profissionais do tipo 5 e assim por diante. Pode haver mais de um profissional do mesmo tipo, o que significa que um paciente será atendido pelo primeiro profissional que se disponibilizar.

Alguns pacientes possuem carteirinha de atendimento prioritário (marcados com a palavra `preferencial` depois do nome). Esses pacientes são inseridos no *começo* de qualquer fila em que entrarem, mesmo que a primeira pessoa da fila também tenha carteirinha de atendimento prioritário.

Restrições

- Nomes são strings delimitadas por aspas duplas e possuem no máximo 50 caracteres ASCII, podendo conter espaços.
- Não há limite para o número de atendimentos que um paciente pode receber em um dia.

Exemplo de entrada

```
"Allegra Magbie" normal 1 2 3 4 5
"Chiquia Sidra Rockwell" normal 5 4 3 2 1
"Teddi Daniele" normal 1 2 4 3 5
"Queenie Broderic Junia" normal 7 7 4 7 7
```

Saída

Todos os pacientes e profissionais já estão no hospital às 08:00, quando começam os atendimentos. Presuma que cada atendimento leva exatamente 10 minutos, já incluídos o tempos de deslocamento e de chamada de pacientes.

O seu programa deverá criar um registro dos horários de saída dos pacientes em um dia. Deve-se imprimir, em ordem, os horários e nomes de cada pessoa que sai do hospital. Se mais de um paciente sai no mesmo horário, a ordem de impressão na saída não importa.

Exemplo de saída

```
08:50 Chiquia Sidra Rockwell
08:50 Allegra Magbie
08:50 Teddi Daniele
08:50 Queenie Broderic Junia
```

CrITÉRIOS

É obrigatório o uso da estrutura de dados deque (*double ended queue*) implementada usando listas ligadas. Um deque é uma generalização de fila que contém operações de inserção e remoção tanto no início quanto do final. Cada uma dessas operações deve ser implementada em tempo constante. Não pode haver vazamento de memória.

Dicas e observações

- No script de correção, a compilação inclui todos os arquivos `.c` no diretório `tarefa06`. Tenha certeza que exatamente um deles possui a função `main()`. Você pode incluir outros arquivos criando arquivos `.h` correspondentes.
- A sua saída pode ser ligeiramente diferente dos arquivos `.res`. As saídas são ordenadas antes de serem comparadas pelo script de correção. Para comparar as saídas uma a uma, você pode usar o comando de Bash `diff -y <(sort testes/sala_de_espera1.res) <(./sala_de_espera <testes/sala_de_espera1.in | sort)`.
- Se dois ou mais pacientes terminarem um atendimento ao mesmo tempo, pode acontecer que eles queiram entrar na fila para a mesma especialidade. Dependendo da ordem em que eles entrarem nessa fila, o horário de saída desses pacientes pode mudar e quem foi inserido depois pode acabar sendo prejudicado, causando um conflito. Por isso, é importante adotar um critério de desempate justo. Neste hospital, o critério de desempate adotado para evitar conflitos é inserir primeiro pacientes que chegaram antes no hospital (de acordo com a ordem do arquivo de entrada). Nesta tarefa, é permitido o uso de funções de ordenação prontas, como o `qsort()`, disponível em `stdlib.h`.
- Você sabia que `scanf()` retorna um int? A documentação pode te ajudar. No Linux, `man scanf` imprime a documentação da função `scanf`. Ela pode ser facilmente encontrada online, também.

Correção

Esta tarefa será corrigida automaticamente e por um monitor. O teste automático será realizado sempre que você realizar um `push`. Para indicar que a sua tarefa está pronta para a correção pelo monitor, remova o arquivo `nao_corrigir.txt` do seu repositório antes de dar `commit`. A nota dependerá do número de casos de teste executados corretamente de acordo com a completude da tarefa:

- Casos de teste sem pacientes prioritários nem conflitos ($\frac{10}{20}$): conceito C
- Casos de teste sem conflitos ($\frac{16}{20}$): conceito B
- Todos os casos de teste ($\frac{20}{20}$): conceito A

No entanto, nota poderá ser descontada por questões de organização, legibilidade ou eficiência. Pense na próxima pessoa que lerá seu código, pode ser você daqui a um mês!