

Enunciado do Projecto 1 - IAED 2018/19

Data de entrega: 04 de Abril de 2019 (17h59m)

LOG alterações

- (22Mar2019) Corrigidos os outputs dos ficheiros de teste. Datas foram normalizadas para a forma ddmmaaaa e tempos para hhmm.
- (22Mar2019) Adicionada nota referente ao máximo de eventos por sala.
- (22Mar2019) Adicionado o comando x.
- (22Mar2019) Clarifica as mensagens de erros, e erros sobrepostos.

1. Introdução

O objectivo deste projeto é o desenvolvimento, em linguagem C, de um sistema de reserva de salas. A interacção com o programa deverá ocorrer através de um conjunto de linhas compostas por uma letra (comando) e um número de argumentos dependente do comando a executar. Os possíveis comandos são listados na Tabela seguinte e indicam as operações a executar.

Comando Acção

a	Adiciona um novo evento
l	lista todos os eventos por ordem cronológica
s	lista todos os eventos de uma dada sala por ordem cronológica
r	apaga um evento
i	altera a hora de início de um evento
t	altera a duração de um evento
m	muda a sala de um evento
A	adiciona um participante a um evento
R	remove um participante de um evento
x	termina o programa

2. Especificação do problema

O objectivo do projecto é ter um sistema de reserva de salas onde podem ser agendadas reuniões/eventos.

Existem várias salas de reuniões onde podem ser agendados eventos.

Cada evento tem:

- uma descrição (uma string com um máximo de 63 caracteres),
- um dia em que se vai realizar (no formato ddmmaaaa),
- um tempo de início (no formato hhmm)
- uma duração (em minutos),
- a sala em que se irá realizar (um inteiro entre 1 e 10),
- uma pessoa responsável pelo evento (uma string com um máximo de 63 caracteres), e
- uma lista de participantes para além do responsável (uma string em que os participantes são separados pelo caracter ':' com um mínimo de 1 e um máximo de 3). Pode assumir que o nome de cada um dos participantes também terá um máximo de 63 caracteres.

Nunca poderão existir eventos sobrepostos numa mesma sala nem responsáveis/participantes que estejam em mais que um evento ao mesmo tempo.

O identificador de eventos é a sua descrição que poderá assumir ser única. Nunca serão criados 2 eventos distintos com a mesma descrição.

Pode assumir que cada evento termina no dia em que começa, ie, não passa para o dia seguinte, e que a sua duração será em minutos (entre 1 e 1440=24horas). Se necessário pode assumir que nenhum evento foi agendado antes de 01012019.

Pode assumir que existem no máximo 10 salas, numeradas de 1 a 10, e que nunca serão agendados mais que 100 eventos por sala ao mesmo tempo.

Nota (22Mar2019): Não viola a hipótese acima a criação de 100 eventos para uma sala, seguida da remoção de um evento, seguida da criação de outro.

Pode assumir que todo o input fornecido respeitará os tipos indicados. Por exemplo, nunca será dada uma sala que seja uma string, um valor negativo, zero, nem um valor superior a 10.

Pode também assumir que as operações mais comuns serão as operações do tipo a, l e s.

3. Dados de Entrada

O programa deverá ler os dados de entrada a partir da linha de comandos e do terminal.

Durante a execução do programa as instruções devem ser lidas do terminal (standard input) na forma de um conjunto de linhas iniciadas por um carácter, que se passa a designar por comando, seguido de um número de informações dependente do comando a executar; o comando e cada uma das informações são separados pelo carácter ':':

Os comandos disponíveis são descritos de seguida. Cada comando indica uma determinada acção que se passa a caracterizar em termos de formato de entrada, formato de saída, e erros a retornar:

- a - Adiciona um novo evento
 - Formato de entrada: a descricao:data:inicio:duracao:sala:responsavel:participantes
 - Formato de saída: NADA (excepto erro)
 - Erros:
 - Impossivel agendar evento <descricao>. Sala<sala> ocupada. no caso de a sala estar ocupada no período em causa;
 - Impossivel agendar evento <descricao>. Participante <participante> tem um evento sobreposto. se a sala estiver disponível mas o responsável ou um dos participantes já tiver um evento sobreposto. Se vários tiverem indisponibilidade deverá retornar todas as indisponibilidades pela ordem responsável, participante1, participante2, participante3.
- l - lista todos os eventos por ordem cronológica do seu início. No caso de existirem vários eventos que se iniciem ao mesmo tempo, deverão ser listados pela ordem crescente do número da sala.
 - Formato de entrada: l
 - Formato de saída: Um par de linhas no formato abaixo, por cada evento existente,

Sala

■

Os participantes deverão ser listados pela ordem em que foram introduzidos.

- Erros: Não aplicável
- s - lista todos os eventos de uma dada sala por ordem cronológica
 - Formato de entrada: s sala
 - Formato de saída: Igual ao formato de saída do comando l
 - Erros: Não aplicável
- r - apaga um evento
 - Formato de entrada: r descricao
 - Formato de saída: NADA (excepto erro)
 - Erros:
 - Evento <descricao> inexistente. no caso de o evento não existir
- i - altera a hora de início de um evento
 - Formato de entrada: i descricao:novo_inicio
 - Formato de saída: NADA (excepto erro)
 - Erros:
 - Evento <descricao> inexistente. no caso de o evento não existir
 - Os mesmos que os erros para o comando a no caso de o evento existir
- t - altera a duração de um evento
 - Formato de entrada: t descricao:nova_duracao
 - Formato de saída: NADA (excepto erro)
 - Erros:
 - Evento <descricao> inexistente. no caso de o evento não existir
 - Os mesmos que os erros para o comando a no caso de o evento existir
- m - muda a sala de um evento

- Formato de entrada: m descricao:nova_sala
 - Formato de saída: NADA (excepto erro)
 - Erros:
 - Evento <descricao> inexistente. no caso de o evento não existir
 - Impossível agendar evento <descricao>. Sala<sala> ocupada. no caso de a sala estar ocupada no período em causa
- A - adiciona um participante a um evento já existente. Se o participante já participar no evento, então não deverá fazer nenhuma acção
 - Formato de entrada: A descricao:participante
 - Formato de saída: NADA (excepto erro)
 - Erros:
 - Evento <descricao> inexistente. no caso de o evento não existir
 - Impossível adicionar participante. Evento <descricao> ja tem 3 participantes. no caso de o evento já ter o máximo de 3 participantes
 - Impossível adicionar participante. Participante <participante> tem um evento sobreposto. no caso de ser possível adicionar participantes mas o participante já tiver um evento sobreposto
- R - remove um participante de um evento já existente. Se o participante não participar no evento, então não deverá fazer nenhuma acção
 - Formato de entrada: R descricao:participante
 - Formato de saída: NADA (excepto erro)
 - Erros:
 - Evento <descricao> inexistente. no caso de o evento não existir
 - Impossível remover participante. Participante <participante> e o unico participante no evento <descricao>. no caso de o participante ser o único participante do evento (para além do responsável).

4. Dados de Saída

O programa deverá escrever no standard output as respostas aos comandos apresentados no standard input. As respostas são igualmente linhas de texto formatadas conforme definido anteriormente neste enunciado. Tenha em atenção o número de espaços entre elementos do seu output, assim como os espaços no final de cada linha. Procure respeitar escrupulosamente as indicações dadas.