

Laboratório em Sala 05

Prof. Arnaldo Moura

Prazo para entrega: 23/05/2016 às 17:59:59

1 AGENDA

Alice tem até 72 horas para organizar todos os seus compromissos e viajar. Desta forma, ela organiza todos os seus compromissos em uma agenda. Mas eventualmente precisa realocá-los ou cancelá-los.

Você fará um programa para ajudar Alice no gerenciamento dos seus compromissos na sua agenda. Para cada compromisso, sabe-se sua descrição e tempo de duração em horas. Você receberá um fluxo do agendamento e cancelamentos dos compromissos de Alice. Desta forma, você irá alocar e desalocar espaços na agenda de acordo com o fluxo recebido. A alocação deve respeitar a seguinte regra: dado a descrição do compromisso e a horário inicial, aloca-se o espaço na agenda a partir da hora inicial ate a hora final de acordo com a duração do compromisso. Por exemplo: seja o compromisso “IrAoDentista” com duração 3 horas e seja o agendamento para este evento “IrAoDentista” 14h, você deverá alocar o horário de 14h às 16h59m ([14h-17h]) na agenda de Alice.

O horário inicial de um agendamento consiste de um inteiro indicando o dia e outro inteiro indicando a hora. Para uma agenda/planejamento com 72 horas, as 24 primeiras horas correspondem ao dia 1, as próximas 24 ao dia 2 e as demais ao dia 3. Desta forma o agendamento de “IrAoDentista” no dia 2 às 14h representa um evento agendado após 38 horas em sua agenda. Alice sempre começa a organizar seus compromissos a partir da 0h do primeiro dia até, no máximo, 72h depois.

Os compromissos podem ser repetidos durante o dia, por exemplo o evento “lanche” com duração de 1 hora. Desta forma, para cancelar um compromisso, você deverá desalocar a primeira ocorrência deste evento.

O fluxo de agendamentos e cancelamentos é sempre consistente. Só existe agendamento de compromissos em horários livres da agenda. Desta forma, para ocorrer o agendamento do compromisso “IrAoDentista”, os horários de 14h as 16h59m devem estar livres. E, só existe cancelamento de compromissos que já foram agendados. Além disso, as durações dos compromissos e o horário inicial do agendamento são sempre inteiros (os minutos são desprezados).

Dado o fluxo, você deverá verificar a configuração resultante da agenda de Alice e responder: 1) quantas horas estão disponíveis? 2) Qual a duração máxima que um compromisso pode ser agendado?

Nesta solução deste problema você deverá implementar um **vetor de estruturas** (vetor para os compromissos) e outro **vetor** para representar as alocações de compromissos em horários da agenda. **Obrigatoriamente**, o compromisso deve ser uma estrutura contendo uma string para sua descrição e um inteiro para a sua duração. Para a agenda, pode ser usado um **vetor** (de inteiros, de strings, ...) em que cada posição do vetor indica qual compromisso está agendado naquela hora. Ainda, cada posição do vetor corresponde a um intervalo de uma hora da seguinte maneira: $[x - y[$ correspondendo ao intervalo xh até $xh59m$, tal que $x - y = 1$. Veja o exemplo a seguir: o intervalo $[14 - 15[$ inicia na $14h$ e termina na $14h59m$.

No susy, em arquivos auxiliares, contém o arquivo **main.c** com implementação de algumas funções para a leitura e impressão de dados. Você poderá editá-lo e submetê-lo.

Entrada:

- Uma linha contendo um inteiro positivo H menor ou igual a 72, que representa a quantidade de horas Alice irá planejar em sua agenda;
- Uma linha contendo um inteiro positivo N menor ou igual a 20, que representa a quantidade de compromissos distintos de Alice;
- N linhas contendo cada compromisso (descrição e duração) envolvido. Onde a descrição contém até 20 caracteres sem espaço e o tamanho é um inteiro positivo menor que 10 representando a quantidade de horas de duração do compromisso;
- Uma linha com um inteiro M indicando a quantidade de registros de agendamentos e cancelamentos;
- M linhas indicando cada registo do agendamento ou cancelamento de um compromisso. As linhas serão no formato $A \text{ descrição } D \ H$ ou $C \text{ descrição}$ onde A representa agendamento, descrição é a descrição do compromisso, D e H representam o dia e a hora que o compromisso será agendado, e C representa cancelamento de um compromisso.

Saída:

- “ x horas disponíveis para compromissos com duração menor ou igual a y ”. Onde x é a quantidade de horas disponíveis na agenda após o fluxo, e y é o tamanho máximo de horas consecutivas disponíveis após o fluxo.

Exemplo:

Entrada:

30 // numero de horas para planejar na agenda

4 // num de compromissos

Dormir 6 // descrição e duração

IrAoDentista 3

escreverArtigo 5

Lanche 1

8 // num de registros de A/C

A Dormir 2 0 //registros => Dormir inicia no dia 2 a 0 hora

A Dormir 1 0

A IrAoDentista 1 14

A Lanche 1 17

A escreverArtigo 1 18

C Lanche

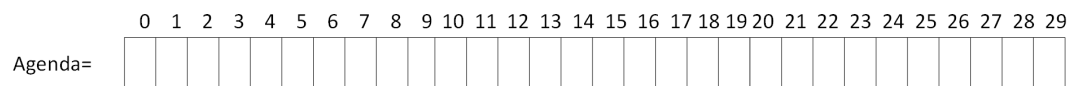
A escreverArtigo 1 8

C escreverArtigo

Saída:

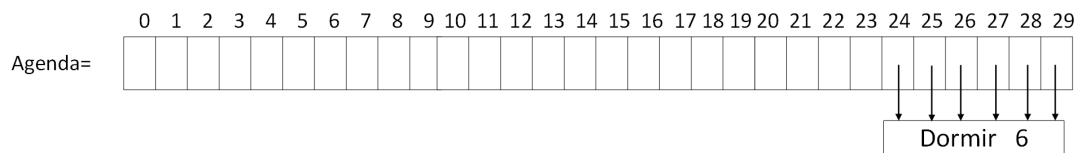
10 horas disponíveis para compromissos com duração menor ou igual a 8

Configuração da agenda após cada agendamento e cancelamento:

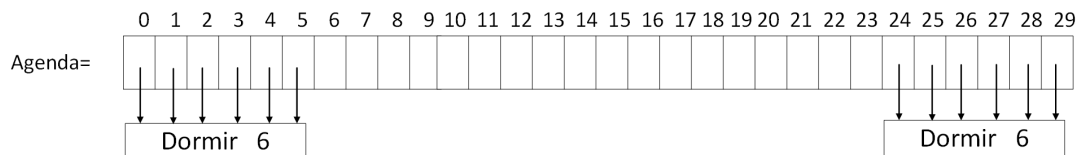


Nesta Agenda, cada posição armazena uma hora. A posição 0 armazenará compromissos de 0h a 0h59m, a posição 1 armazena de 1h a 1h59min, e assim sucessivamente.

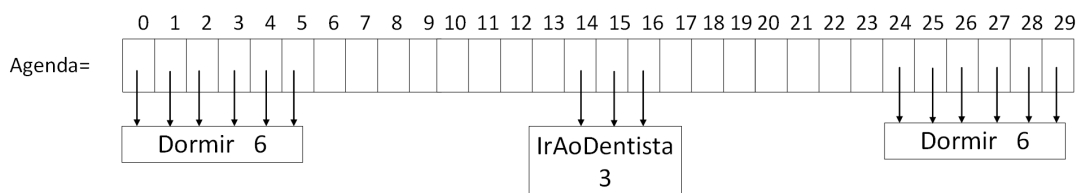
A Dormir 2 0



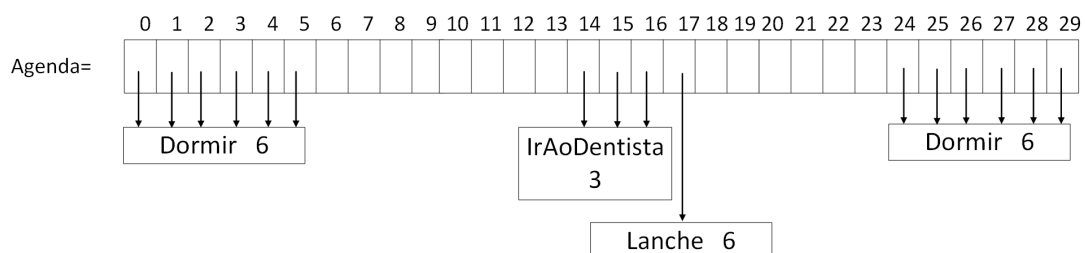
A Dormir 1 0



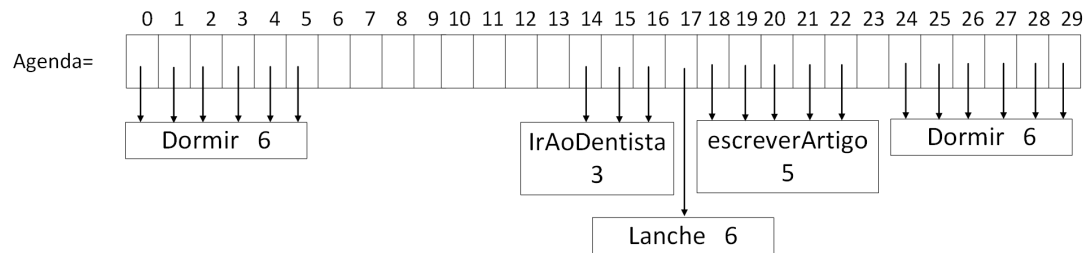
A IrAoDentista 1 14



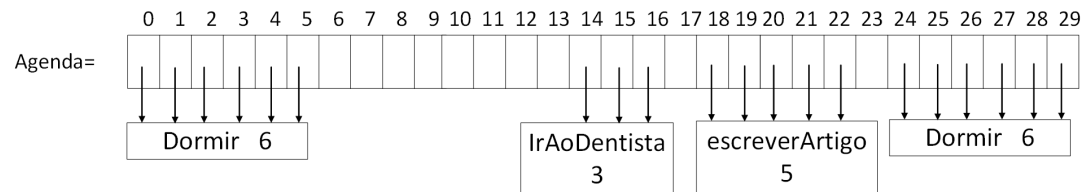
A Lanche 1 17



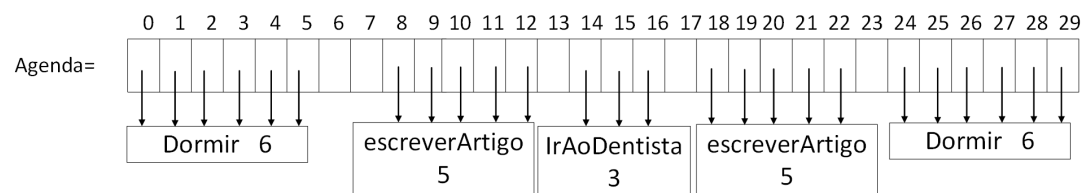
A escreverArtigo 1 18



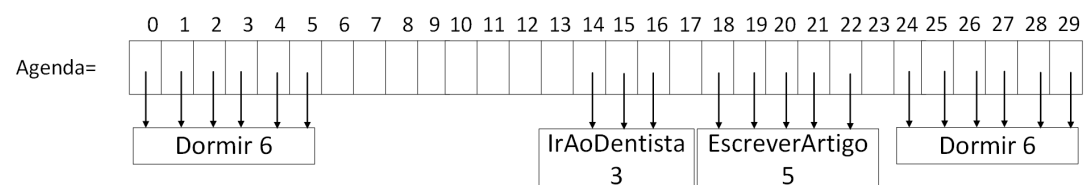
C Lanche



A escreverArtigo 1 8



C escreverArtigo



• Observações

- Você poderá usar a função `strcmp` da biblioteca `string.h` para comparar strings. A função retorna 0 se as strings são iguais.

```
int strcmp ( const char * str1, const char * str2 );
```

- Você poderá utilizar funções para dividir o problema em subproblemas. Sugestões de subtarefas que podem ser implementadas por funções:

- * Agendar um compromisso na agenda
- * Cancelar um compromisso na agenda
- * Contar horas disponíveis na agenda
- * Calcular a duração máxima que um compromisso pode ter para ser alocado na agenda, isto é, maior slot com horas vagas consecutivas

- Tarefas/funções já implementadas no arquivo **main.c**, disponível no **susy**:

- * Ler os compromissos e armazenar no vetor de estruturas

- * Ler registros de agendamento e cancelamento
- * Imprimir agenda e compromissos (deve ser usada apenas para teste, essas impressões não devem fazer parte da saída)
- ***É proibido o uso de variáveis globais***
- *O programa deve ser submetido em C (labSala05.c)*
- *Faça comentários e indentação do seu código*
- *O aluno pode assumir que todas as linhas da entrada terminam com o fim-de-linha*
- *Todas as linhas da saída devem terminar com o fim-de-linha*
- *O número máximo de submissões é 15*
- *O comando de compilação será:*
gcc -std=c99 -pedantic -Wall -lm labSala05.c -o labSala05
- *O comando de execução será:*
./labSala05 ou ./labSala05 <arq00.in