

Empresa de Transportes “Semprarrolar” – escalonamento de condutores –

INTRODUÇÃO

A empresa de transportes **Semprarrolar** pretende desenvolver um programa informático para fazer o escalonamento dos seus condutores. A empresa é obrigada a assegurar o transporte de passageiros num conjunto de **linhas** (percursos).

Uma **linha** é caracterizada pelos seguintes atributos: número identificador único; frequência de circulação dos autocarros (pode ser diferente de linha para linha mas não varia com o dia da semana); uma lista com a sequência das paragens que a constituem; uma lista com os tempos de viagem entre paragens. Uma **linha** funciona de modo bidirecional (da paragem inicial para a paragem final e de volta à paragem inicial). Uma **paragem** é identificada por um nome único e pode pertencer a várias linhas.

Cada **condutor** da empresa é caracterizado por um nome, um número identificador único, um número de horas consecutivas que pode conduzir por dia (um turno), um número máximo de horas que pode conduzir por semana e um número mínimo de horas de descanso obrigatório entre turnos.

O objetivo último do trabalho prático é desenvolver um programa que faz, para uma semana o escalonamento dos condutores dos autocarros para todas as linhas que a empresa tem de assegurar, respeitando os limites máximos de horas de condução e horas de descanso obrigatórias. Neste primeiro trabalho serão apenas desenvolvidas as funcionalidades básicas. Estas funcionalidades serão complementadas no segundo trabalho onde o desenvolvimento da aplicação completa será concluído.

OBJETIVOS

Este trabalho prático tem como objetivo desenvolver um programa para preparar a aplicação de escalonamento dos condutores da frota da empresa **Semprarrolar**. O programa a desenvolver deve definir novos tipos de dados adequados à aplicação, criar as estruturas de dados adequadas, ler e gravar informação para ficheiros. Não é objetivo desenvolver um programa completo que faça o escalonamento dos condutores; esse será o objetivo do trabalho prático nº 2, usando um conjunto de regras que serão indicadas oportunamente.

Ao realizar este trabalho, os estudantes terão oportunidade de aplicar os conhecimentos de programação em C++ adquiridos até ao momento, nomeadamente, quanto a:

- desenvolvimento de interfaces simples;
- tratamento de entradas do teclado inválidas;
- formatação de saídas;
- leitura e escrita em ficheiros de texto;
- uso de estruturas de controlo de um programa (seleção e repetição);
- uso de diversos tipos de estruturas de dados (*strings*, *arrays/vectors*, *structs*, *files*);
- definição de novos tipos de dados apropriados para a aplicação ;
- uso de funções, com passagem de parâmetros por valor e por referência.

DESENVOLVIMENTO DO PROGRAMA

Definição de novos tipos de dados

Utilizando a facilidade de definição de estruturas (**struct**) e de definição de novos tipos de dados (**typedef**), disponíveis na linguagem, comece por definir dois novos tipos de dados: para uma **linha** e para um **condutor**.

Nota: Apesar de o 1.º trabalho prático não ser classificado, é condição necessária para a obtenção de frequência o desenvolvimento, entrega e apresentação (quando solicitada) deste trabalho. O trabalho será realizado individualmente.

Funcionamento geral

O programa deve começar por ler o ficheiro de **linhas** e o ficheiro de **condutores**. O ficheiro de linhas tem, para cada linha, a seguinte informação: um identificador numérico único; a frequência dos autocarros na linha; a sequência de paragens; e uma lista com os tempos de viagem entre paragens. A frequência, expressa em minutos, indica o intervalo de passagem de um autocarro em cada paragem. Uma frequência com valor 15 (minutos) indica que em qualquer paragem daquela linha passa um autocarro de 15 em 15 minutos. O ficheiro de condutores armazena, para cada condutor, a seguinte informação: o identificador numérico único do condutor; o nome do condutor; o número máximo de horas do seu turno; o número máximo de horas que pode conduzir por semana; e o número mínimo de horas de descanso entre turnos. O formato dos ficheiros será especificado adiante.

Ficheiro de linhas e de condutores

Qualquer um dos ficheiros utilizados no programa (linhas e condutores) são ficheiros de texto que podem ser criados com um editor de texto simples como, por exemplo, o *Notepad* do *Windows*. O programa deve começar por perguntar ao utilizador qual o nome dos dois ficheiros a usar, terminando com um erro caso algum dos ficheiros indicados não exista.

Apresenta-se a seguir um possível conteúdo de um **ficheiro de linhas**.

<i>Exemplo de um ficheiro de linhas (inventadas):</i>	
201 ; 20 ;	Paranhos, Monte dos Burgos, ISEP, Hospital de Sao Joao; 10, 20, 45
207 ; 10 ;	Aliados, Carmo, Galiza, Rotunda da Boavista; 5, 20, 20
505 ; 30 ;	Matosinhos, Sete Bicas, Rotunda dos Produtos Estrela, Senhora da Hora; 40, 30, 45
506 ; 15 ;	Ribeira, Alfandega, Fluvial, Foz; 10, 25, 55

Cada uma das linhas do ficheiro contém a seguinte informação acerca de cada linha: o número identificador único; frequência de circulação dos autocarros na linha (em minutos); uma lista com a sequência das paragens que a constituem; uma lista com os tempos (em minutos) de viagem entre paragens. Os diferentes tipos de dados estão separados entre si por ";" e os elementos que fazem parte de uma lista estão separados entre si por ",".

Apresenta-se a seguir um possível conteúdo de um **ficheiro de condutores**.

<i>Exemplo de um ficheiro de condutores:</i>	
1 ; Antonio Manuel ; 4 ; 36 ; 2	
3 ; Joaquim Belmiro ; 8 ; 48 ; 4	

Cada uma das linhas do ficheiro tem a seguinte informação acerca de cada condutor: um identificador numérico único; um nome; o número de horas consecutivas que pode conduzir por dia (um turno); o número máximo de horas que pode conduzir por semana; e o número mínimo de horas de descanso obrigatório entre turnos. Os itens estão separados entre si por ";" .

Assuma que o transporte se faz apenas durante o dia e defina como constantes as horas de início e de fim dos serviços diários.

Funcionalidades a implementar

A parte da aplicação que deve implementar neste trabalho inclui as seguintes funcionalidades:

- Ler e guardar a informação de linhas e condutores armazenada em ficheiros.
- Gerir as linhas e condutores: criar, alterar e remover uma linha ou um condutor.
- Gerar e visualizar de modo formatado tabelas com horários de uma paragem.
- Gerar e visualizar de modo formatado o horário de uma linha.
- Visualizar o trabalho atribuído a um condutor.
- Inquirir sobre quais as linhas que incluem determinada paragem.
- Listar condutores disponíveis.
- Calcular e visualizar um percurso e tempos entre duas quaisquer paragens indicadas pelo utilizador.
- Calcular, para uma linha especificada, quantos condutores são necessários (assumindo turnos com um número de horas fixo).

MELHORIAS OPCIONAIS

Apresentam-se a seguir alguns desafios para desenvolvimentos adicionais:

- Introduzir informação sobre autocarros e distâncias entre paragens. Os autocarros têm autonomia limitada e tempos e locais de abastecimento.
- Considerar dois tipos de linhas: circulares e não circulares.
- Melhorar as estruturas de dados de modo a tornar a aplicação mais eficiente.

NOTAS SOBRE O DESENVOLVIMENTO

Funcionamento geral do programa

O funcionamento geral do programa é o seguinte:

- Ler o conteúdo dos ficheiros de linhas, condutores e parâmetros da aplicação e guardar numa estrutura de dados adequada.
- Começar por exibir um menu onde aparecem as opções que implementam as funcionalidades: gestão de linhas; gestão de condutores; visualização de informação (horários, linhas e percursos).
- No final, gravar (se necessário) a informação das linhas e dos condutores.

Devem ser tomadas as precauções necessárias para evitar que o programa deixe de funcionar corretamente devido a entradas inválidas do utilizador, nomeadamente, valores fora da gama admissível.

Escrita do código

Na escrita do código devem ser respeitadas as indicações dadas nas aulas, nomeadamente, em relação aos seguintes aspetos:

- Escolha das estruturas de dados mais adequadas para representar os dados do programa.
- Escolha adequada dos identificadores de tipos, variáveis e funções.
- Estrutura modular do código.
- Comentários ao código.

ENTREGA DO TRABALHO

- Criar uma pasta com o nome **TxEyyyyyyyyyy**, em que **x** representa o número da turma e **yyyyyyyyyy** representa o número do estudante (sem as letras **up** ou **ei**); por exemplo, **T2E201600007**, para o estudante com o número 201600007 da turma 2) e copiar para lá o código-fonte do programa (apenas os ficheiros com a extensão .cpp e .h, se existirem). Incluir também um ficheiro **ReadMe.txt** (em formato de texto simples) indicando o estado de desenvolvimento do trabalho, isto é, se foram cumpridos todos os objetivos ou, caso contrário, quais os que não foram cumpridos, ou ainda que melhorias foram implementadas, se for esse o caso.
- Compactar o conteúdo desta pasta num ficheiro **Gxx.zip** ou **Gxx.rar** e submeter este ficheiro na página da unidade curricular de Programação, no Moodle da FEUP. Não serão aceites entregas por outras vias.
- Data limite para a entrega: **18/abr/2017 (às 23:55h)**.