

1. Analise com atenção a definição da classe **graphStl** que contem a implementação de uma estrutura do tipo grafo representada por uma lista de adjacências, bem como as classes **graphEdge** (ramo) e **graphVertex** (vértice). Faça a implementação na classe **GraphStlPath** dos seguintes métodos:
  - a) Visita em largura, iterativo
  - b) Visita em profundidade, recursivo
  - c) Todos os caminhos simples (sem vértices repetidos) entre dois vértices  $v_i$  e  $v_j$ , podendo  $v_i=v_j$
  - d) Caminho mínimo entre dois vértices
  
2. Para cada exposição um museu define vários trajetos de visita às suas salas, tendo em conta vários critérios, como por exemplo, tempo de visita, períodos da exposição, etc. Cada exposição pode normalmente ter início em várias salas e entre duas salas podem existir vários trajetos possíveis. Cada uma das salas caracteriza-se por um número e pelo seu tempo médio de visita. Considera-se que o tempo de deslocação entre salas é desprezável. Supondo que a rede de salas definida para uma exposição é representada por um grafo usando a classe **GraphStlPath**, defina o grafo e respetivas classes TV e TR que melhor se adaptam para responder às alíneas a seguir indicadas:
  - a) Determinar o tempo necessário para visitar todas as salas de uma exposição.
  - b) Apresentar todos os percursos possíveis entre duas salas da exposição, cujo tempo de visita é inferior a um determinado valor.
  - c) Apresentar o percurso completo (que envolve todas as salas da exposição), e respetivo tempo de visita, com início numa determinada sala.
  
3. Considere um grafo para representar países e relações de fronteira entre os mesmos, de tal modo que um país só tem ligação com outro se os países forem fronteiriços. Cada país é representado pela sua capital e continente. As ligações entre os países incluem a distância em km entre as respetivas capitais. Supondo que este mapa é representado por um grafo não dirigido usando a classe **GraphStlPath**:
  - a) Defina a classe Mapa e respetivas classes TV e TE indicando apenas os atributos e a assinatura dos métodos necessários para gerir esta rede de países.

Acrescente à classe Mapa os métodos para:

- b) Verificar se o grafo possui um circuito Euleriano, ou seja, se partindo de uma capital é possível percorrer todas as ligações entre os países, sem repetição destas.
- c) Identificar as duas capitais mais próximas no mapa, indicando as duas capitais e a distância em Km entre as mesmas.

4. Considere a utilização da classe GraphStlPath na representação do plano de um curso em que algumas unidades curriculares (UCs) apresentam precedências entre si e as UCs são representadas pela classe UnidCurric.

```
class UnidCurric{
private:
    string nome;
    int ECTS;
    int ano;

public:
    UnidCurric ();
    UnidCurric (const UnidCurric& uc) ;
    ~UnidCurric ();

    void setNome(string n) ;
    void setECTS(int ec) ;
    void setAno(int a) ;
    string getNome() const ;
    int getECTS() const ;
    int getAno() const ;
    ...
};
```

- a) Apresente a definição da classe não template **PlanoCurso** (não é necessário indicar os métodos da classe).
- b) Implemente o método UCSemPreced, que apresenta as UCs sem precedências
- c) Implemente o método UCMaisPreced, que apresenta a(s) UC(s) com mais precedências
- d) Implemente um método que apresenta uma possível ordem para a realização de todas as UCs do curso obedecendo ao plano de precedências do mesmo.