

Resolva cada exercício em folhas separadas

1. Considere as seguintes classes relativas a países e à definição de uma viagem.

```
class pais {
private:
    string nome, continente;
    set<string> idiomas;

public:
    pais() ;
    pais(const pais& a) ;
    ~pais() ;
    ...
} ;

class viagem {
    queue<pais> paises;

public:
    viagem() ;
    viagem(const viagem& t) ;
    ~viagem() ;
    ...
} ;
```

Acrescente às classes pais e viagem **TODOS** os métodos necessários para realizar as seguintes operações:

- Adicionar à viagem um país a visitar, mantendo a informação organizada por ordem alfabética do nome do país.
- Indicar quantos idiomas diferentes se poderão falar numa viagem.

2. Considere o seguinte método

```
int misterio (vector<int> A, int n)
{
    if (n==1)
        return A[0];
    else
    {
        int k=n/2;
        for (size_t i=0; i<k; i++)
        {
            if (n%2==0)
                A[i] = teste(A[i], A[i+k]);
            else
                A[i] = teste(A[i], A[i+k+1]);
        }
        if (n/2 != 0)
            A[0] = teste(A[0], A[(n/2)]);

        return misterio(A, k);
    }
}

int teste (int j, int k)
{
    if (j>k)
        return j;
    else
        return k;
}
```



- Diga o que faz o método misterio.
- Faça a análise a complexidade temporal  $T(n)$  do código quando invocado o método misterio (A,9). Justifique.

---

Resolva cada exercício em folhas separadas

3. Um turista pretende planear uma visita a vários países, efetuando apenas viagens por avião. Considere a classe Pais apresentada no exercício 1.
- a) Defina a classe NÃO TEMPLATE **MapaAereo** de forma a descrever um **grafo dirigido** que represente a rede de países e o preço de cada viagem de avião entre eles (**não é necessário indicar os métodos da classe**).
  - b) Implemente um método que dado um determinado continente, apresenta ao turista todos os países que poderá visitar desse mesmo continente.
  - c) Para ajudar o turista a planear uma viagem que inclua apenas 3 países, implemente um método que dado o país de origem do turista, lhe indique quantas opções diferentes dispõe.
4. Defina na classe template `tree<TN>` um método que valida se um elemento X existe na árvore.
5. Considere um vetor com os seguintes elementos: 11,75,2,23,14,4. Verifique se corresponde a um heap (prioridade ordem decrescente). Construa a árvore correspondente a este heap. Justifique.