

1. Pretende-se que defina uma classe para implementar um sistema de reposição das requisições dos exemplares de uma biblioteca. Assuma que todos exemplares da biblioteca são identificados pelo seu ISBN e estão guardados numa lista ligada, ordenados por tema. As devoluções são guardadas numa pilha. Periodicamente os exemplares da pilha de devoluções são repostos na lista de obras da Biblioteca, por actualização do número de exemplares disponíveis. Admita que existe sempre na biblioteca um exemplar físico de cada obra para consulta.

Considere a classe **Ficha** que guarda informação relativa aos exemplares disponíveis e às requisições da Biblioteca.

```
class Ficha
{
    private:
        string isbn;
        string tema;
        string autor;
        int numexemp; //nº de exemplares disponíveis ou
                     //nº exemplares requisitados

    public:
        Ficha();
        Ficha (string t, string a, int n);
        Ficha (const Ficha& f);
        ~Ficha () ;

        string getISBN() const ;
        string getTema() const ;
        string getAutor() const ;
        int getNExemp() const ;

        void setTema (string mat) ;
        void setAutor (string nom) ;
        void setNExemp (int num) ;

        Ficha& operator = (const Ficha& f) ;
        bool operator == (const Ficha& f) const ;
        void escreve (ostream& out) const ;
};
```

- Defina a classe **BibRequis** para implementar o sistema de reposição das requisições desta biblioteca (não é necessário indicar os métodos da classe).
- Implemente o método **replacement** que devolve à Biblioteca todas as obras da pilha de requisições.
- Faça um método que apresenta o(s) livro(s) mais requisitado(s) da pilha de requisições.

2.

```
void exemplo (int a[], int li, int ls)
{
    if (li < ls)
    {
        processa (a,li,ls) ;
        ls=ls/2 ;
        exemplo (a,li,ls) ;
    }
}
```

```
void processa (int a[], int liminf, int limsup)
{
    int i=liminf ;
    int j=limsup-1 ;

    while (i < j)
    {
        int temp=a[i] ;
        a[i]=a[j] ;
        a[j]=temp ;
        i++ ;
        j-- ;
    }
}
```

- a) Explique o que faz o código acima e apresente o resultado aplicado ao vector $a[6]=\{1,2,3,4,5,6\}$, $li=0$, $ls=6$, exemplo (a,0,6)
- b) Analise a sua complexidade temporal, notação BigOh. Justifique.



3. Considere um parque de diversões com várias actividades de entretenimento para as quais se sabe o seu código de identificação, tempo médio de espera e tempo de duração. Para além disto sabe-se também o tempo de percurso entre os pavilhões das várias actividades. Supondo que este parque é representado por um grafo dirigido da classe template ListAdjGrafo<TV,TR>:

- a) Apresente a definição das classes necessárias à modelação deste problema (não é necessário indicar os métodos das classes).
- b) Apresente o **caminho mínimo** entre uma actividade inicial e uma actividade final, e indique o tempo total do caminho encontrado separado nas suas várias componentes: tempo total de duração das actividades, tempo total de espera e tempo total de percurso.

4. Acrescente ao template ArvBinPesq<T> um método para encontrar o **maior elemento** de uma árvore binária de pesquisa, da forma mais eficiente possível, e indique a complexidade temporal, notação BigOh, do método desenvolvido.