Algoritmos e Estruturas de Dados, Prova de Avaliação, 6 de Julho de 2001

Duração 2 horas, com consulta apenas de material próprio trazido para o efeito

Nome (legível): _____

Pergunta 1 [5 valores]

Considere o seguinte programa em C++:

```
1.  #include <iostream.h>
2.  template <class T> class Variavel {
3.   private:
4.    char nome[20];
5.    T valor;
6.   public:
7.    template <class T2> Variavel(const char *nm, const T2 & val)
8.    { nome = nm; Write(val); }
9.    const char GetNome() const
10.   { return nome; }
11.   template <class T2> T2 Read(T2 & val) const
12.   { val = (T2) valor; }
13.   template <class T2> void Write(const T2 & val) const
14.   { valor = (T) val; }
15. };
16. main()
17. { Variavel<int> x ("x", 10.5);
18.   Variavel<double> * y = new Variavel("y", 10.5);
19.   double d;
20.   cout << x.GetNome() << "=" << x->Read(d) << "\n";
21.   cout << y.GetNome() << "=" << y->Read(& d) << "\n";
22.   return 0;
23. }
```

Identifique claramente e corrija os erros contidos neste programa (para cada erro, basta indicar a linha que contém o erro, descrever o erro e rescrever a linha corrigida). Apresente ainda os resultados da execução do programa corrigido.

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Nome (legível): _____

Pergunta 2 **[6 valores]**

Escreva uma função `RemoveDups` a acrescentar à classe `LinearList` estudada nesta disciplina, para eliminar da lista todos os valores repetidos.

```
template<class T>
class LinearList {
   public:
      LinearList(int MaxListSize = 10); // constructor
      ~LinearList() {delete [] element;} // destructor
      bool IsEmpty() const {return length == 0;}
      int Length() const {return length;}
      bool Find(int k, T& x) const;
         // return the k'th element of list in x
      int Search(const T& x) const;
         // return position of x
      LinearList<T>& Delete(int k, T& x);
         // delete k'th element and return in x
      LinearList<T>& Insert(int k, const T& x);
         // insert x just after k'th element
      void Output(ostream& out) const;
   private:
      int length;
      int MaxSize;
      T *element; // dynamic 1D array
};
```

Nome (legível): _____

Pergunta 3 **[5 valores]**

Usando a classe `BinarySearchTree` estudada nesta disciplina, escreva um pequeno programa que lê uma série de números inteiros positivos do *standard input*, não ordenados e com repetições, e escreve no *standard output* os mesmos números ordenados sem repetições. Considere que a série de números de entrada termina com 0.

```
template <class T>
class BinaryNode {
   BinaryNode(const T & theElement, BinaryNode * lt, BinaryNode * rt)
      : element(theElement), left(lt), right(rt) { }
   T element; BinaryNode * left; BinaryNode * right;
};
template <class T>
class BinarySearchTree {
  public:
    explicit BinarySearchTree(const T & notFound);
    BinarySearchTree(const BinarySearchTree & rhs);
    const T & findMin() const; const T & findMax() const; //...
    void insert(const T & x);  void remove(const T & x); //...
  private:
    BinaryNode<T> *root; ...
    const T & elementAt(BinaryNode<T>  *t) const;
    void insert(const T & x, BinaryNode<T> * & t) const;
    void remove(const T & x, BinaryNode<T> * & t) const;
    BinaryNode<T> * findMin(BinaryNode<T> * t) const;
    BinaryNode<T> * findMax(BinaryNode< T> * t) const; //...
};
```

Nome (legível): _____

Pergunta 4 **[4 valores]**

Pretende-se desenvolver um programa para efectuar a "junção" de um ficheiro de funcionários com um ficheiro de departamentos, conforme ilustra o seguinte exemplo:

Ficheiro de funcionários (um dos ficheiros de entrada):

```
codigo_func | nome_func | codigo_dep
1 | José Silva | 1
2 | Maria Rebelo | 3
3 | Álvaro Campos | 1
4 | Rui Pereira | 2
```

Ficheiro de departamentos (outro dos ficheiros de entrada):

```
codigo_dep | nome_dep
1 | Logística
2 | Contabilidade
```

Ficheiro de resultado (junção de funcionários com departamentos):

```
codigo_func | nome_func | codigo_dep | nome_dep
1 | José Silva | 1 | Logística
2 | Maria Rebelo | 3 | ?
3 | Álvaro Campos | 1 | Logística
4 | Rui Pereira | 2 | Contabilidade
```

O ficheiro de resultado é semelhante ao ficheiro de funcionários, acrescido de um campo com o nome do departamento correspondente ao código do departamento. Quando o código do departamento não consta no ficheiro de departamentos, é escrito "?" no lugar do nome. Todos os ficheiros estão ordenados pelo 1º campo e são ficheiros de texto. O ficheiro de funcionários pode ser muito grande.

Diga que estruturas de dados e algoritmos estudados nesta disciplina (ou construídos a partir deles) seriam adequados para resolver este problema eficientemente (em tempo e espaço) e apresente uma breve justificação das escolhas efectuadas. Indique, justificando, qual seria a complexidade temporal e espacial desse programa.

Fim.