



Faculdade de Engenharia da Universidade do Porto
 Licenciatura em Engenharia Electrotécnica e de Computadores
 Algoritmos e Estruturas de Dados
 Ano lectivo de 2000/2001

Prova de Avaliação, duração 2 horas, com consulta

20 de Junho de 2001

Nome: _____

Pergunta 1 [5 valores]

Considere o seguinte programa em C++:

```

1:  #include <iostream.h>
2:  template <class T> class Poligono // T: int, double, etc.
3:  {private:
4:      const int numMaxPontos;
5:      int numActualPontos;
6:      T *xx, *yy;
7:  public:
8:      void Poligono(int numMaxPontos)
9:          { this->numMaxPontos = numMaxPontos; numActualPontos = 0;
10:           xx = new T [numMaxPontos]; yy = new T [numMaxPontos];
11:         }
12:      ~Poligono() { delete [] xx; delete [] yy; }
13:      Poligono & adicionaPonto(T x, T y)
14:          { xx[numActualPontos] = x; yy[numActualPontos] = y; return *this; }
15:      void imprime() const
16:          { for (int i = 0; i < numActualPontos; i++)
17:             a.      cout << "(" << xx[i] << "," << yy[i] << ")\\n";
18:           }
19:  main()
20:  { Poligono<int> p(4);
21:    p.adicionaPonto(1,1).adicionaPonto(2,2).adicionaPonto(3,3);
22:    p.imprime();
23:    return 0;
24:  }

```

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

Pergunta 2 [6 valores]

Considere o tipo de dados abstracto lista e a sua implementação baseada em cadeia na classe `Chain`, estudados nesta disciplina.

Adicione a esta classe um membro-função pública `ContainsChain` que receba como parâmetro uma referência a outra cadeia, por exemplo `c2`, e retorna `true` no caso da instância de `Chain` sobre a qual o membro-função é invocado conter `c2` como sub-cadeia e `false` no caso contrário.

```
template <class T>
class ChainNode {
    friend Chain<T>;
private:
    T data;
    ChainNode<T> *link;
};
template<class T>
class Chain {
public:
    Chain() {first = 0;}
    ~Chain();
    bool IsEmpty() const {return first == 0;}
    int Length() const;
    bool Find(int k, T& x) const;
    int Search(const T& x) const;
    Chain<T>& Delete(int k, T& x);
    Chain<T>& Insert(int k, const T& x);
    void Output(ostream& out) const;
private:
    ChainNode<T> *first, // pointer to first node
};
```

Pergunta 3 [5 valores]

Escreva um membro-função `isSearchTree` a acrescentar à classe `BinaryTree` estudada nesta disciplina, que verifica se a árvore binária é uma árvore binária de pesquisa, devolvendo `true` em caso afirmativo e `false` em caso negativo.

```
template <class T>
class BinaryTreeNode {
public:
    BinaryTreeNode() {LeftChild = RightChild = 0;}
    BinaryTreeNode(const T& e) {data = e; LeftChild = RightChild = 0;}
    BinaryTreeNode(const T& e, BinaryTreeNode *l, BinaryTreeNode *r)
        {data = e; LeftChild = l; RightChild = r;}
private:
    T data;
    BinaryTreeNode<T> *LeftChild, // left subtree
                   *RightChild; // right subtree
};
template<class T>
class BinaryTree {
public:
    BinaryTree() {root = 0;};
// (...)
private:
    BinaryTreeNode<T> *root;
// (...)
};
```

Pergunta 4 [4 valores]

Pretende-se desenvolver um programa que verifica se um dicionário de sinónimos tem a propriedade reflexiva. O dicionário de sinónimos encontra-se guardado num ficheiro de texto. Cada linha do ficheiro contém um termo seguido de uma lista de zero ou mais sinónimos. O ficheiro encontra-se ordenado por ordem alfabética dos termos. Exemplo:

```
alvo   branco claro
branco alvo claro
claro  alvo
escuro negro
negro  escuro
```

Diz-se que o dicionário tem a propriedade reflexiva se, sempre que o termo $t1$ está na lista de sinónimos do termo $t2$, então o termo $t2$ está na lista de sinónimos do termo $t1$. Por exemplo, o dicionário apresentado acima não tem a propriedade reflexiva porque "claro" está na lista de sinónimos de "branco", mas o inverso não se verifica. No entanto, o programa apenas tem de indicar se o dicionário tem ou não a propriedade reflexiva, não tendo que o justificar.

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