Faculdade de Engenharia da Universidade do Porto
LICENCIATURA EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES
ALGORITMOS E ESTRUTURAS DE DADOS - 2000/2001

2º MINI-TESTE DURAÇÃO: 30 minutos SEM CONSULTA TURMA: 2E14 DATA: 22/5/2001
Responda nos rectângulos em branco!

Nome do aluno: ***Resolução*** **Turma:**

1. Preencha uma errata relativa ao seguinte pedaço de código em C++ (Nota: as linhas estão numeradas para mais fácil referência):

```
1:  class Conta {
2:   public:
3:    Conta(long saldo = 0) {saldo = saldo; numero = ++ult_numero;}
4:    long getSaldo() const {return saldo;}
5:    static int getNumero() const {return numero;}
6:    void putSaldo(long novo_saldo) const {saldo = novo_saldo;}
7:   private:
8:    const int numero;       // número da conta, único e imutável
9:    long saldo;             // saldo da conta, em escudos
10:   static int ult_numero;  // contador para numeração automática
11: }
12: int Conta.ult_numero = 0;
```

Na linha	Onde está	Devia estar
3	`) {saldo = saldo; numero = ++ult_numero;}`	`) : numero = (++ult_numero);} {this.saldo = saldo;}`
5	`static int getNumero()`	`int getNumero()`
6	`const {saldo = novo_`	`{saldo = novo_`
11	`}`	`};`
12	`Conta.ult_numero = 0;`	`Conta::ult_numero = 0;`

2. Assinale com um V (verdadeiro) ou um F (falso) cada uma das afirmações seguintes (Nota: uma resposta errada desconta uma certa):

a) ***V*** Em C++, é possível copiar objectos da mesma classe com o operador de atribuição.

b) ***F*** Em C++, um membro privado (`private`) de uma classe pode ser acedido por membros-função de classes derivadas.

c) ***V*** Em C++, uma função virtual pura é uma função que não tem implementação na classe em que está definida.

d) ***V*** Numa lista linear representada em *array*, a inserção de um elemento na posição *k* pode ser efectuada em tempo *O((n-k)s)*, em que *n* é o nº de elementos na lista e *s* é o tamanho de cada elemento.

e) ***F*** Numa lista linear em que a operação mais frequente é a selecção de um elemento que se encontra numa posição arbitrária, a representação mais adequada é a representação em cadeia.

3. Escreva uma classe `Intervalo`, para representar intervalos de números reais (do tipo `double`), com os seguintes membros:

- dados privados do tipo `double`, denominados `inicio` e `fim`, para guardar o valor inicial e final do intervalo;
- um construtor que aceita como argumentos os valores inicial e final do intervalo;
- funções públicas `getInicio` e `getFim`, para obter o valor inicial e final do intervalo, respectivamente;
- uma função pública `contem`, que aceita como argumento um valor do tipo `double`, e devolve um booleano que indica se o intervalo contém esse valor;

```
class Intervalo {
  public:
    Intervalo (double in, double fi) {inicio= in; fim= fi;}
    double getInicio () const {return inicio;}
    double getFim () const {return fim;}
    boolean contem (double valor) const
                    {return (valor >= inicio && valor <= fim);}
  private:
    double inicio;
    double fim;
};
```