

# 2ª Prova de Algoritmos e Estruturas de Dados I

## 16/05/2012

### Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.  
R: A compreensão do enunciado faz parte da prova.
- P: Se eu consultar algum material próprio ou de algum colega, o que acontecerá comigo?  
R: A prova é individual e sem consulta. Qualquer tentativa de fraude acarretará abertura de processo administrativo na UFPR.
- P: Posso entregar a prova a lápis?  
R: Prova é um documento, portanto deve ser entregue a caneta.
- P: O que será corrigido?  
R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, a indentação, uso equilibrado de comentários no código e, evidentemente, a clareza. A modularidade, correto uso de funções e procedimentos, incluindo passagem de parâmetros e bom uso de variáveis locais e globais serão especialmente observados.

### 1. Questão 1 (35 pontos)

Fazer um programa em Pascal que calcule a média dos dígitos de um número inteiro com qualquer quantidade de dígitos. Use o tipo `integer` para o número de entrada.

Exemplo:

`entrada: 19825`

`saida: 5.00`

(que é o valor da expressão  $(1+9+8+2+5)/5$ , com duas casas depois da vírgula)

### 2. Questão 2 (20 pontos)

Fazer apenas uma função em Pascal chamada de `menor_primo_depois_de` que recebe como parâmetro apenas um número inteiro positivo. A função retorna o menor número primo que também é maior do que o parâmetro recebido pela função. Por exemplo, a ativação da função `menor_primo_depois_de` no fragmento de código do programa principal abaixo retorna o valor 23 para ser somado com 3.1416 e, no final, o valor resultante 26.1416 é atribuído à variável `x`:

```
x := menor_primo_depois_de(20) + 3.1416;
```

Um outro exemplo de ativação da função `menor_primo_depois_de` seria o que segue:

```
writeln(menor_primo_depois_de(3));
```

O resultado impresso na saída padrão seria:

5

### 3. Questão 3 (30 pontos)

Fazer um programa completo em Pascal capaz de ler dados de entrada onde cada linha contém 6 (seis) valores numéricos correspondentes a 3 (três) pares ordenados do espaço cartesiano. Os 3 pares ordenados também podem ser vistos como 3 pontos desse espaço cartesiano que sempre formam as coordenadas dos vértices de um triângulo (o programa não precisa verificar isso). A quantidade de linhas do arquivo é desconhecida. A última linha do arquivo contém 6 (seis) valores zero e é a **única** cujos valores não formam um triângulo. Para cada linha de dados lida, o programa deve calcular e imprimir os valores dos comprimentos dos lados do triângulo. Um exemplo de execução seria assim:

```

./triangulos
1.3 2.1 6.4 3.5 3.2 1.8
5.29 1.92 3.62
0.0 0.0 0.0 3.0 4.0 0.0
3.00 4.00 5.00
0.0 0.0 0.0 0.0 0.0 0.0

```

Dica: a distância entre dois pontos cujas coordenadas são  $(x_1, y_1)$  e  $(x_2, y_2)$ , também chamada de distância Euclidiana, é dada pela expressão:

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

4. Questão 4 (15 pontos)

Qual é a saída que o programa abaixo produz?

```

program misterio;
var
    n1, n2, n3 : integer;

function f (n1, n2 : integer): integer;
var
    a : integer;
begin
    a:= n1;
    while a < n2 do
        a:= a * 2;
    f:= a;
end;

procedure p (var a: integer; var b: integer; c: integer);
var
    n1 : integer;
begin
    n1:= (a + b) mod c;
    if a < b then
        a:= a + n1
    else
        b:= b + n1;
end;

begin
    n1:= 10;
    n2:= 20;
    n3:= 4;
    p(n1, n2, n3);
    writeln(f(n1, n2));
end.

```