

Exame de Introdução à Computação

Exame de recurso, 4 de Julho 2002, 9:00-11:00 (Duração: 2 horas)
Universidade do Algarve



- Escreve o teu nome, nº de aluno e curso em todas as folhas que entregares.
- Não é permitido falar com os colegas durante o exame. Se o fizeres, terás a prova anulada. Desliga o telemóvel.
- Caso optes por desistir, escreve “Desisto”, assina e entrega a prova ao docente.
- O exame tem 7 perguntas e a cotação de cada aparece entre parêntesis.
- Calculadoras são proibidas.
- Faz letra legível. Usa a linguagem PASCAL para os programas.
- Boa sorte!

Pergunta 1 (1 valor)

Explica a função de um compilador.

Pergunta 2 (3 valores)

O programa a seguir mostra no ecrã ‘Olá mundo’. Muda o programa de forma que o output vai para o ficheiro com nome ‘OLA.TXT’.

```
PROGRAM Ola;
begin
  Writeln('Ola mundo');
end.
```

Pergunta 3 (3 valores)

Faz um procedimento com nome Soma com parâmetro n (inteiro) que calcula (e mostra no ecrã) o resultado da soma

$$1 \times 2 + 2 \times 3 + 3 \times 4 + \dots + (n-1) \times n$$

Pergunta 4 (3 valores)

Qual será o output do seguinte programa? Justifica a tua resposta.

```
PROGRAM Pergunta4;

FUNCTION FunN(n: integer): integer;
begin
  if n=0 then FunN := 0
  else
    begin
      if (n MOD 2) = 0 then
        FunN := 1 + FunN(n-1)
      else
        FunN := FunN(n-1);
    end;
end;

begin
  Writeln(FunN(21));
end.
```

Pergunta 5 (7 valores, 3+2+2)

Este programa é sobre quadrados mágicos. Um quadrado mágico é uma matriz quadrada preenchida com números 1 até n^2 onde a soma dos elementos das linhas, colunas e as diagonais são iguais. A figura ao lado mostra um exemplo com $n = 3$ e com somas iguais a 15.

6	1	8	15
7	5	3	15
2	9	4	15
15	15	15	15

Assume que um quadrado mágico fica no array $A[1..n, 1..n]$.

- Escreve uma função `somalinha` com argumento k que retorna a soma da linha k . Escreve também funções similares para as colunas e as diagonais. A última função deve ter um argumento que determina qual a diagonal que deve ser verificada (1=*top-left to bottom-right*, 0=*top-right to bottom-left*).
- Escreve uma função `Verifica` que verifica se todas as somas de todas as linhas e colunas são iguais. A função deve retornar `TRUE` se todas as somas são iguais, `FALSE` caso contrário.
- Escreve uma *procedure* que verifica se todos os números pertencentes ao intervalo $1..n^2$ são usados no quadrado mágico. O procedimento deve escrever o resultado no ecrã.

Pergunta 6 (2 valores)

Parece que os dois fragmentos de código a seguir sempre fazem a mesma coisa.

<pre>if CondC then begin InstrS; InstrT; end else begin InstrS; InstrU; end; end;</pre>	<pre>InstrS; if CondC then begin InstrT; end else begin InstrU; end; end;</pre>
---	---

Dá um exemplo para a condição `CondC` e as instruções `InstrS`, `InstrT` e `InstrU` que mostra a diferença em estrutura (o mesmo estado no início dá estados diferentes no fim ou gera output diferente)

Pergunta 7 (1 valor)

Explique a diferença entre uma variável global e uma variável local.

Funções e procedimentos de PASCAL

<i>Função</i>	<i>descrição</i>	<i>argumento</i>	<i>result</i>	<i>exemplos</i>
Abs	Valor absoluto do argumento. Argumento pode ser real ou inteiro Abs retornará o mesmo tipo.	real ou integer	real ou integer	Abs(-23.2) = 23.1 Abs(12.3) = 12.3 Abs(-10) = 10
Cos	Cosine do argumento. Argumento em radians (2π rad = 360°)	real	real	Cos(1.0) = 0.5403
Sin	Sine do argumento. Argumento em radians (2π rad = 360°)	real	real	Sin(1.0) = 0.8415
ArcTan	Inverso tangent do argumento	real	real	ArcTan(1.0) = $\pi/4$
Exp	Exponent (e^x) do argumento	real	real	Exp(1.0) = 2.718
Ln	Logaritmo (nepereano) do argumento	real (>0)	real	Ln(10.0) = 2.303
Odd	Determine se o argumento é impar	integer	boolean	Odd(3) = TRUE
Round	Arredondamento do argumento para o inteiro mais perto	real	integer	Round(3.4) = 3 Round(3.5) = 4
Int	Arredondamento do argumento para o inteiro abaixo	real	real	Int(3.99) = 3.00
Frac	Retorno a parte do número depois o ponto decimal	real	real	Frac(3.99) = 0.99
Trunc	Arredondamento do argumento para o inteiro abaixo	real	integer	Trunc(3.99) = 3
Sqrt	Raiz do argumento	real (>0)	real	Sqrt(3.0) = 1.732
Sqr	Quadrado do argumento	real	real	Sqr(2.0) = 4.0
Random	Gera número aleatório	sem ou integer	real ou integer	Random = 0.0234 Random(10) = 3
Randomize	Randomizar o gerador dos números aleatórios			
Assign	Atribuir o nome do ficheiro	TEXT, string		Assign(f, 'TEXT.TXT')
Rewrite	Abrir o ficheiro para escrever	TEXT		Rewrite(f)
Reset	Abrir o ficheiro para ler	TEXT		Reset(f)
Close	Fechar o ficheiro	TEXT		Close(f)

1) A compiler is to translate the text understandable by humans to code understandable by the computer.

2)

```
PROGRAM Ola;
Var f: text;
begin
  Assign(f, 'OLA.TXT');
  Rewrite(f);
  WriteLn(f, 'Ola Mundo');
  Close(f);
end.
```

3)

```
PROCEDURE Soma(n: integer);
Var i, som: integer;
begin
  som := 0;
  for i := 1 to n-1 do
    som := som + i * (i+1);
  writeln(som);
end;
```

4) The function counts the number of pair numbers in the series 1 to n in a recursive method. Therefore, FunN(21) counts the numbers 20, 18, 16, 14, 12, 10, 8, 6, 4, 2. These are 10 numbers and so, the output will be 10.

```
FunN(21)
= FunN(20)
= 1+FunN(19)
= 1+FunN(18)
= 1+1+FunN(17)
= 1+1+FunN(16)
= 1+1+1+FunN(15)
= 1+1+1+FunN(14)
= 1+1+1+1+FunN(13)
= 1+1+1+1+FunN(12)
= 1+1+1+1+1+FunN(11)
= 1+1+1+1+1+FunN(10)
= 1+1+1+1+1+1+FunN(9)
= 1+1+1+1+1+1+FunN(8)
= 1+1+1+1+1+1+1+FunN(7)
= 1+1+1+1+1+1+1+FunN(6)
= 1+1+1+1+1+1+1+1+FunN(5)
= 1+1+1+1+1+1+1+1+FunN(4)
= 1+1+1+1+1+1+1+1+1+FunN(3)
= 1+1+1+1+1+1+1+1+1+FunN(2)
= 1+1+1+1+1+1+1+1+1+1+FunN(1)
= 1+1+1+1+1+1+1+1+1+1+1+FunN(0)
= 1+1+1+1+1+1+1+1+1+1+1+0
= 10
```

5a)

```
FUNCTION SomaLinha(k: integer): integer;
Var i, soma: integer;
begin
  soma := 0;
  for i := 1 to n do
```

```

        soma := soma + A[k, i];
    SomaLinha := soma;
end;
FUNCTION SomaColuna(k: integer): integer;
Var i, soma: integer;
begin
    soma := 0;
    for i := 1 to n do
        soma := soma + A[i, k];
    SomaColuna := soma;
end;
FUNCTION SomaDiagonal(k: integer): integer;
Var i, soma: integer;
begin
    soma := 0;
    for i := 1 to n do
        if k=1 then
            soma := soma + A[i, i]
        else
            soma := soma + A[i, n-i+1];
    SomaDiagonal := soma;
end;

```

5b)

```

FUNCTION Verifica: boolean;
var k, soma: integer;
begin
    Verifica := TRUE;
    soma := SomaLinha(1);
    for k := 2 to n do
        if SomaLinha(k) <> soma then
            Verifica := FALSE;
    for k := 1 to n do
        if SomaColuna(k) <> soma then
            Verifica := FALSE;
    if SomaDiagonal(1) <> soma then
        Verifica := FALSE;
    if SomaDiagonal(0) <> soma then
        Verifiac := FALSE;
end;

```

5c)

```

PROCEDURE AllNumbers;
Var i,j: integer;
    used: boolean;
begin
    for i := 1 to n*n do
        begin
            write(i);
            used := FALSE;
            for j := 1 to n do
                for k := 1 to n do
                    if A[j,k] = i then

```

```

        used := TRUE;
    if used then
        writeln(': USED')
    else
        writeln(': NOT USED');
    end;
end;

```

6)

for example: a=0;

<pre> if a=1 then begin a := 1; WriteLn('Benfica'); end else begin a := 1; WriteLn('Sporting'); end; end; </pre>	<pre> a := 1; if a=1 then begin WriteLn('Benfica'); end else begin WriteLn('Sporting'); end; end; </pre>
--	--

7) A local variable only has a meaning / has a value / can be used inside a module (function or procedure or unit). A global variable has a scope throughout the entire program.