UNIVERSIDADE FERNANDO PESSOA

Disciplina: Programação **Ano:** 1º (época de finalistas) **Ano Lectivo:** 2002/2003
Exame: sem consulta; **Duração**: 2h 00m **Data:** 18 de Setembro de 2003
Licenciaturas: Engenharias (Ambiente, Civil, Comunicação, Informática)

1) Escreva um programa que leia do utilizador uma palavra e verifique se ela é simétrica.

Exemplo 1:

Palavra (p)? sacas
sacas é simétrica

Exemplo 2:

Palavra (p)? pessoa
pessoa não é simétrica

1.1) Escreva o algoritmo em pseudo-código.
1.2) Escreva o respectivo algoritmo usando um fluxograma.
1.3) Faça o seguimento e teste (rastreio) do algoritmo para a palavra 'ana'.
1.4) Modifique o pseudo-código de 1.1 de modo a que este tome a forma de uma função que aceita uma string e devolve um valor booleano correspondente à informação sobre se a string passada é simétrica (verdadeiro) ou não (falso). Construa um programa em pseudo-código que depois de pedir ao utilizador para introduzir uma dada frase ou string, inspecciona se existe alguma substring dessa string de tamanho igual ou superior a 2 caracteres que seja ela própria simétrica. Para tal o programa deve utilizar a função pedida no inicio desta alínea. O programa deve listar no ecrã todas as substrings simétricas da string lida do utilizador.

2) Dada uma matriz `m` de números inteiros com tamanho NxN (N é um valor constante), escreva um algoritmo em pseudo-código que calcule a média de todos os valores `m[i,j]` da matriz em que a quantidade `(i+j)` é um número par.

3) Dada a seguinte declaração:

```
type
 aluno = record
  nome: string[60]; idade: integer; media_curso: integer;
 end;
var
 alunos: array[1..100] of aluno;
```

Escreva um procedimento que calcule e apresente no ecrã a média das médias de curso de todos os alunos do vector `alunos` que têm ou idade menor do que `idade_1` ou idade maior do que `idade_2` lidos do utilizador e não se chamam 'Santos'. Considere que o vector se encontra totalmente preenchido.

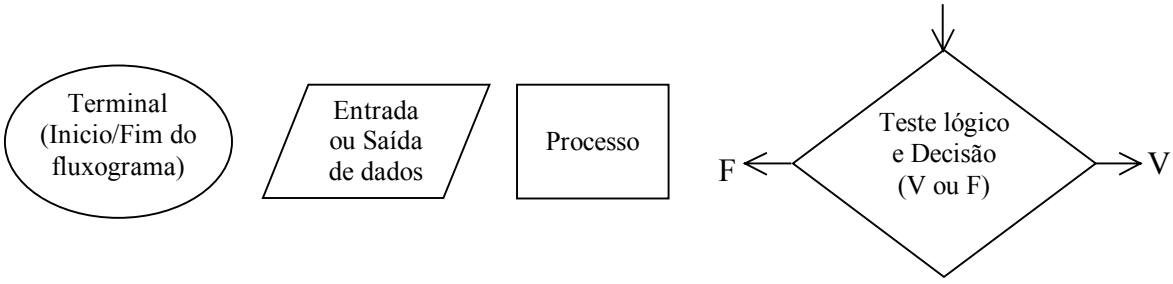
[Cotação em valores: 1.1) 3; 1.2) 2; 1.3) 2,5; 1.4) 2,5; 2) 4,5; 3) 5,5]

Sugestões de notação para descrição dos seus algoritmos ao longo do teste:

Pseudo-código:

- Leia(...) e LeiaLinha(...)
- Escreva(...) e EscrevaLinha(...)
- Para <*expressão_inicial*> Até <*expressão_final*> Faça <*bloco*>
- Se <*condição*> Então <*bloco*> Senão <*bloco*>
- Repita <*bloco*> AtéQue <*condição*>
- Enquanto <*condição*> Faça <*bloco*>
- Caso <*expressão*> Seja <*caso_1*> : <*bloco*> .... <*caso_n*> : <*bloco*> Senão <*bloco*> FimCaso
- <*operador de atribuição*>: ←
- <*bloco*>: Inicio .... Fim

Fluxogramas:



(Nota1: poderá, principalmente na descrição textual do algoritmo, usar outra notação com a qual esteja mais à vontade como por exemplo o Pascal ou outra. Não se preocupe demasiado com a sintaxe mas não deixe de ser claro na descrição do seu algoritmo
Nota2: poderá usar os operadores relacionais, lógicos e aritméticos normais assim como algumas funções matemáticas mais vulgares)

Vectores e Cadeias de Caracteres:

Considere os vectores e matrizes como sendo de tamanho fixo
Relativamente às variáveis do tipo *string* considere o seguinte:

- Pode consultar/alterar o valor de um caracter dentro da string através da operação normal de acesso a elementos de um vector
- Operador de concatenação de strings: +
- Função que devolve o comprimento da string: length(*s*)
- Função para copiar/extrair substring de uma string: copy (*s, index_ini, count*)
- Função para determinar posição de uma substring dentro de uma string: pos(*substr, s*)
- Conversão de string para um valor numérico: val(*s, v, error_code*)
- Conversão de um valor numérico para string: str(*x, s*)