



Departamento de Matemática da Universidade de Coimbra

(P) Métodos de Programação I 2005/2006 Exame Final 2h30m

Leia atentamente o enunciado de cada pergunta antes de iniciar a sua resolução.

Sempre que use tipos não-elementares deve ser apresentada a respectiva declaração de tipo que usar.

Exercício 1 (7, 0)

Um número primo é um **natural** que apenas é **divisível** por ele mesmo e pela unidade. **Sabe-se** que, dado **p primo**, o número de Mersenne, $2^p - 1$ pode ser **também** primo (ou não).

- a) **Escreva** e implemente **em** Pascal, uma função que recebe um inteiro, m , e devolve uma resposta booleana, que **será true** se e só se m é um número primo.
- b) Elabore e implemente um procedimento que, dado um número inteiro **positivo**, n e, usando a **função** anterior, caso n seja primo, **calcula** o **número** de **Mersenne**, $2^n - 1$ e, se este for **também** primo, escreve no **output** o número n , seguido de **alguns** espaços e **m** branco e, **finalmente**, o valor $2^n - 1$.
- c) Usando **as alíneas** anteriores, escreva **um** programa em **Pascal** para:

- 1. ler e armazenar uma sequência de números inteiros positivos numa tabela unidimensional denominada **Num**;
- 2. usando o procedimento da alínea (b), escrever para o **output**, em forma de grelha, todos os elementos de **Num** que resultem em Primos de **Mersenne**, bem como o **respectivo** valor de primo de Mersenne. A título de exemplo, se os primeiros valores **armazenados** em **Num** forem 2, 12, 5, 4, 3, ..., a **grelha** deve começar **assim**:

Numero	Primo de Mersenne
2	3
5	31
3	7

. . .

Não se esqueça de escrever **todo** o programa, declarando todos **os** novos tipos ou sub-programas que **achar** necessários para o correcto funcionamento do seu programa. Apenas no caso **das funções** das alíneas anteriores, pode usar **apenas** o cabeçalho no **sítio correcto** (ou seja, **declarando-os apenas** no sítio onde **são** precisos), mas pode evitar **declarar** o **bloco**, uma vez que já o fez antes.

Exercício 2 (4, 0) Tenha ainda em **atenção** o exercício 1 e o tipo de **array** **ai** declarado.

- a) Pense agora no seguinte: a grelha do exercício anterior vai ficar **"desordenada"**, com valores maiores e menores misturados, sendo, por exemplo, **difícil verificar** que primos **faltam** pelo meio (nem todo o primo p dá origem a um primo de **Mersenne**). Para evitar isto, antes de usar o **módulo** que escreve a grelha, **dever-se-ia ordenar**, por ordem crescente do seu **valor**, os elementos de **Num**.

Escreva um procedimento para, dado um vector do tipo usado no grupo anterior e respectiva dimensão útil, efectuar este **ordenamento**, usando o método da **Borbulhagem** (*Bubblesort*).

- b) Descreva os pontos principais de construção de um **algoritmo** recorrente.
- c) **Explique** como pode transformar o algoritmo da **Borbulhagem** num **algoritmo** recorrente e escreva o correspondente procedimento.

Exercício 3 (2, 0) Considere as seguintes declarações de tipos:

```

type
  Digito = 0..9;
  Nat     = array [0..MAX] of Digito;

```

Suponha que usamos o tipo *Nat* acima para guardar a **representação** decimal de números naturais. Por exemplo, a representação do número $n = 25059$ num **vector** do tipo *Nat* será:

0	1	2	3	4	5	...	MAX
9	5	0	5	2	0	...	0

Dados dois números **naturais** através dos seus representantes em *Nat*, $A[0..MAX]$ e $B[0..MAX]$, descreva e implemente um **algoritmo** para **verificar** se B é **capicua** de A.

bigskip

Exercício 4 (4, 00)

- a) Explique que diferenças **existem** entre uma instrução de **while-do**, uma instrução de repeat-**until** e uma **instrução** de for-to-do.
- b) Em que situações se deve **escolher** qual dos ciclos.
- c) Elabore um programa capaz de receber do teclado uma **sucessão** de números inteiros negativos cujo **fim** deverá ser assinalado com uma sentinela adequada. Pretende-se determinar o maior elemento da sucessão e o número de **vezes** que ocorre na **sucessão**.

Escolha o valor da **sentinela** mais adequado, justificando a sua **escolha**, e escreva um programa em Pascal capaz de responder à **especificação anterior**.

Exercício 5 (3, 00)

Um computador **manipula** informação e os "**átomos**" da **informação** são números e caracteres. Explique com pormenor:

- a) Explique como está, organizada a memória de um computador.
- b) Explique, **pormenorizadamente** e **usando um exemplo**, como armazena a memória de um computador informação numérica do tipo inteiro.