

ÍNDICE DOS PROGRAMAS

Programa 3.1 - Programa que implementa o algoritmo de soma (versão 1).....	134
Programa 3.2 - Programa que implementa o algoritmo de soma (versão 2).....	137
Programa 3.3 - Programa 3.2 optimizado.....	138
Programa 3.4 - Soma da secção 3.3.2.3, em algoritmo (cópia do Programa 3.2, para facilidade de comparação) e em instruções de linguagem <i>assembly</i>	160
Programa 3.5 - Implementação do algoritmo de contagem de <i>bits</i> a 1.....	164
Programa 3.6 - Implementação do semáforo simples.....	175
Programa 3.7 - Implementação do semáforo com botão para peões	176
Programa 4.1 - Instruções do Programa 3.5, que implementa a contagem de <i>bits</i> a 1 no PEPE-8	185
Programa 4.2 - Se o PEPE-8 tivesse três registos, A, B e C, o Programa 4.1 poderia ter este aspecto, mais simples e compacto	186
Programa 4.3 - Exemplo de acesso à memória com índice variável (em elementos): soma de todos os elementos de uma tabela, começando pelo último	225
Programa 4.4 - Exemplo de acesso à memória com índice variável (em <i>bytes</i>): soma de todos os elementos de uma tabela, começando pelo primeiro	226
Programa 4.5 - Exemplo de acesso à memória sem índice (com endereçamento linear): soma de todos os elementos de uma tabela.....	227
Programa 4.6 - Exemplo de acesso à memória com índice constante	229
Programa 4.7 - Exemplo de acesso à memória em 8 <i>bits</i> : cópia de uma cadeia de caracteres com conversão para maiúsculas	233
Programa 4.8 - Exemplo de acesso misto à memória (em 8 <i>bits</i> e 16 <i>bits</i>): cópia de uma cadeia de caracteres ASCII com conversão para <i>Unicode</i>	235
Programa 4.9 - Programa gerador da série de Fibonacci.....	240
Programa 4.10 - Programa para somar dois números de 8 <i>bits</i> num processador de 4 <i>bits</i>	241
Programa 4.11 - Cálculo do factorial (de 6, neste exemplo). O programa pode terminar por erro de excesso se N for grande.....	242
Programa 4.12 - Programa para determinar se um dado número é primo	243
Programa 4.13 - Factorização de um número.....	244
Programa 4.14 - Exemplo de utilização de uma expressão booleana com OR.....	249

Programa 4.15 - Exemplo de utilização de uma expressão booleana com AND que não pode ser implementada com uma simples instrução AND.....	250
Programa 4.16 - Cópia de uma cadeia de caracteres com conversão para maiúsculas (mesmo que as letras originais já o sejam).....	253
Programa 4.17 - Contagem do número de letras maiúsculas com a instrução BIT	254
Programa 4.18 - Exemplo de aplicação de uma máscara AND.....	256
Programa 4.19 - Geração de uma cadeia de caracteres com os dígitos de um número, usando máscaras OR.....	257
Programa 4.20 - Cifra dos caracteres de uma cadeia usando uma máscara XOR.....	260
Programa 4.21 - Contagem do número de <i>bits</i> a 1 num registo com uma instrução de deslocamento	265
Programa 4.22 - Compactação de dois dígitos em BCD num só <i>byte</i> a partir de dois dígitos em ASCII, usando deslocamentos e máscaras.....	266
Programa 5.1 - Programa em linguagem C para ordenar uma sequência de números usando o método de ordenação por bolha (<i>bubble sort</i>)	281
Programa 5.2 - Programa em C para ordenar uma sequência de números usando o método de ordenação por bolha (<i>bubble sort</i>) e apontadores para aceder ao vector	297
Programa 5.3 - Programa em linguagem C, estruturado em funções, para ordenar uma sequência de números usando o método de ordenação por bolha (<i>bubble sort</i>).....	309
Programa 5.4 - Variante do Programa 5.3, com uma alternativa à função <i>ordenaBolha</i> , usando apontadores em vez do acesso a um vector com índice (<i>main</i> e <i>ordena</i> não têm qualquer alteração)	315
Programa 5.5 - Programa simples, com o único fim de ilustrar a guarda/recuperação de registos pelas rotinas	338
Programa 5.6 - Exemplo de acesso a uma tabela com elementos de 16 <i>bits</i> , declarada e inicializada com a directiva <i>WORD</i>	365
Programa 5.7 - Alternativa ao Programa 5.6 com declaração da tabela com a directiva <i>TABLE</i> e inicialização dos elementos por instruções.....	366
Programa 5.8 - Exemplo de acesso a uma tabela com elementos de 8 <i>bits</i> , declarada e inicializada com a directiva <i>STRING</i>	367
Programa 5.9 - Reformulação do Programa 4.6 para exemplificar o acesso a uma tabela bidimensional com elementos de 16 <i>bits</i> (encarada como uma sequência de tabelas monodimensionais).....	371
Programa 5.10 - Exemplo de acesso a uma tabela bidimensional com elementos de 16 <i>bits</i> (encarada como uma só tabela monodimensional linearizada).....	372
Programa 5.11 - Exemplo de acesso a uma tabela bidimensional com elementos de 8 <i>bits</i>	373
Programa 5.12 - Exemplo de utilização de uma tabela de apontadores para estruturas de dados de dimensão variável.....	375

Programa 5.13 - Exemplo de utilização de uma tabela de apontadores para rotinas.....	377
Programa 5.14 - Variante do Programa 5.2 com alocação dinâmica de variáveis no montão.....	380
Programa 5.15 - Exemplo de utilização de uma lista ligada.....	389
Programa 6.1 - Programa que implementa o controlo da intensidade luminosa da lâmpada	474
Programa 6.2 - Espera activa.....	508
Programa 6.3 - Espera não activa (por interrupções)	510
Programa 6.4 - Exemplo de rotina de transferência de dados de um periférico para memória	511
Programa 7.1 - Sequência de instruções para exemplificar o processamento em estágios.....	606
Programa 7.2 - Programação com processos (multiprogramação), com recurso a uma rotina de despacho invocada por interrupções de fatia de tempo	671
Programa 7.3 - Utilização de um trinco lógico para impor exclusão mútua aos processos na execução de uma secção crítica	675
Programa 7.4 - Funções de gestão de semáforos N-ários	677
Programa 7.5 - Programação com processos em regime de programação cooperativa. Este exemplo implementa a mesma funcionalidade que o Programa 7.2.....	682
Programa B.1 - Exemplo de programação do CREPE	724