

## Tratamento de sequências

1. Analise as sub-rotinas e descreva a operação realizada por cada uma delas.

a) 

```
rot_a PROC uses edi apt: ptr byte
      mov edi, apt
      mov ecx, -1
      cld
      mov al, 0
      repnz scasb
      sub edi, apt
      dec edi
      mov eax, edi
      ret
rot_a endp
```

b) Assuma que o último elemento de vect nunca é 20H (código ASCII de espaço).

```
rot_b PROC USES edi vect: PTR BYTE, n: BYTE
      mov edi, vect
      mov al, 20h
      mov cl, n
      xor ebx, ebx
      movzx ecx, cl
@@:   repnz scasb
      jnz @F
      inc ebx
      jmp @B
@@:   mov eax, ebx
      ret
rot_b ENDP
```

2. [Exame de 13-06-2011] Considere a seguinte sequência de instruções:

```
      cld
@@:   lodsw
      inc ax
      stosw
      loop @B
```

Considerando que os registos usados pelas várias instruções foram devidamente inicializados com o objetivo de incrementar os N elementos de 16 bits de uma sequência, após a execução:

- A. EDI possui o valor inicial mais  $4 \times N$ ;
- B. ESI possui o valor inicial menos  $2 \times N$ ;
- C. EDI possui o valor inicial mais N;
- D. ESI = EDI.

3. Considere a seguinte sub-rotina:

```
ROT_X PROC USES EDI EBX AP1:ptr BYTE, Cval:BYTE, N:DWORD
    MOV AL, Cval
    MOV EDI, AP1
    MOV ECX, N
    CLD
    MOV EBX, 0
@@:   INC EDI          ; *
    CMP AL, [EDI-1]    ; *
    LOOPNZ @B          ; *
    JECXZ @F
    INC EBX
    JMP @B
@@:   JNE @F
    INC EBX
@@:   MOV EAX, EBX
    RET
ROT_X ENDP
```

- a) Descreva o funcionamento da sub-rotina, justificando convenientemente a resposta.
  - b) Os microprocessadores da família IA-32 têm instruções próprias, do tipo `REPxx`, para executar ciclos e tratar *buffers*. Com base nesse tipo de instruções, indique como poderia simplificar o fragmento de programa assinalado por `*`.
4. Escrever a sub-rotina `LCASE` que converte todas as letras maiúsculas de um *buffer* de texto para minúsculas. (Assumir que é usada a representação ASCII.)
5. Escrever as sub-rotinas para tratamento de cadeia de caracteres terminadas por zero indicadas abaixo. Cada uma deve verificar se os parâmetros são coerentes e retornar um código que indique se o processamento teve sucesso (0=sucesso, 0≠erro).
- a) `ESQ`: altera uma cadeia de caracteres de forma a manter apenas os `N` caracteres da esquerda.
  - b) `DIR`: altera uma cadeia de caracteres de forma a manter apenas os `N` caracteres da direita.
  - c) `MEIO`: altera uma cadeia de caracteres de forma a omitir os `N` caracteres da direita e os `M` caracteres da esquerda.
6. Escrever as sub-rotinas `LTRIM` e `RTRIM` para remover espaços (código 20H) e caracteres `TAB` (09H), `CR` (0DH) e `LF` (0AH) situados, respetivamente, no extremo esquerdo ou direito de uma cadeia de caracteres. Assumir a representação em código ASCII.
7. Escrever a sub-rotina `BUSCA` que determina, num *buffer* de texto terminado pelo carácter de código 0, a posição da primeira ocorrência de uma sequência de caracteres colocada num segundo *buffer* (também terminado pelo carácter de código 0). A sub-rotina retorna no registo `EAX` a posição da primeira ocorrência ou -1 se não existir nenhuma.

8. Pretende-se determinar o número de ocorrências (frequência) de caracteres num texto.

- a) Para tal, escreva a sub-rotina HIST que aceita um apontador para uma cadeia de caracteres e um apontador para um vetor de 256 DWORDS com valor inicial nulo. Cada elemento do vetor é indexado pelo código ASCII do carácter correspondente. Após execução da sub-rotina, cada entrada do vetor de frequências (histograma) contém o número de vezes que o carácter correspondente ocorre na cadeia de caracteres.
- b) Escrever uma sub-rotina que coloca no monitor uma linha horizontal com  $N$  caracteres #.
- c) Escrever um programa que usa as sub-rotinas anteriores para apresentar graficamente o número de ocorrências de todas as letras minúsculas do texto seguinte.

The first ray of light which illumines the gloom, and converts into a dazzling brilliancy that obscurity in which the earlier history of the public career of the immortal Pickwick would appear to be involved, is derived from the perusal of the following entry in the Transactions of the Pickwick Club, which the editor of these papers feels the highest pleasure in laying before his readers, as a proof of the careful attention, indefatigable assiduity, and nice discrimination, with which his search among the multifarious documents confided to him has been conducted.

Charles Dickens, *The Pickwick Papers*

9. Pretende-se escrever uma sub-rotina para reposicionar um bloco de caracteres numa cadeia de caracteres. Considere que o bloco é constituído por  $N$  caracteres a partir do  $K$ -ésimo carácter da sequência e que o deslocamento a efetuar é de  $D$  caracteres. Caso  $D$  seja positivo o deslocamento é feito para a direita, isto é, os  $N$  caracteres passam a ocupar posições a partir de  $K+D$ , caso  $D$  seja negativo o deslocamento ocorre para a esquerda. Os caracteres originais das posições alteradas são movidos no sentido inverso ao deslocamento do bloco de caracteres.

10. Escrever uma sub-rotina que determina qual dos elementos de um conjunto de caracteres ocorre mais vezes num texto. O conjunto é especificado por uma cadeia de caracteres, sem caracteres repetidos, terminada pelo valor 0. O texto é também terminado por 0. O resultado da sub-rotina (em EAX) é o carácter que ocorre mais vezes no texto; caso exista mais que um carácter nessas condições, deve ser escolhido o que aparece em último lugar na especificação do conjunto.

Exemplo:      Conjunto: aeiou      Texto: alma minha gentil, que te partiste  
Resultado: e

A sub-rotina tem o seguinte protótipo:

```
detect_max proto conjunto:ptr byte, texto:ptr byte
```

**11.** Um texto é constituído por frases e as frases por palavras. A separação das frases é efetuada pelo carácter 'ponto' enquanto que a separação das palavras é feita pelo carácter 'espaço'.

- a) Escrever a sub-rotina `NUM_FRASES` que retorna o número de frases do texto. A sub-rotina tem o seguinte protótipo:

`NUM_FRASES PROTO BUF: PTR BYTE`

- b) Escrever a sub-rotina `NUM_PALAVRAS` que calcula o número de palavras da N-ésima frase do texto. Assuma que as frases são contadas a partir de 1 e caso a frase não exista a sub-rotina deve retornar 0. A sub-rotina tem o seguinte protótipo:

`NUM_PALAVRAS PROTO BUF: PTR BYTE, N: BYTE`

**12.** Nas alíneas seguintes pretende-se escrever sub-rotinas que implementem operações envolvendo matrizes de números inteiros com 16 bits. Do ponto de vista de armazenamento, uma matriz pode considerar-se como um vetor formado pelas sucessivas linhas da matriz. Assim, uma matriz com N linhas e M colunas é uma sequência de N vetores com M elementos, ou seja, é uma sequência de  $N \times M$  elementos. Por exemplo, o elemento da linha de índice 2 e coluna de índice 3 de uma matriz de `WORDS matW` ( $N \times M$ ) pode ser acedido por:

```
.code
mov ebx, offset matW
add ebx, 2*M
mov esi, 3
mov ax, [ebx + esi*type matW]
```

- a) Escrever a sub-rotina `somaMat` que calcula a soma resultante de  $A+B$ , sendo A e B matrizes  $N \times M$ . Os parâmetros da sub-rotina são: *offset* do vetor que representa A, *offset* do vetor que representa B, *offset* do vetor que representa a soma, número de linhas e número de colunas das matrizes.
- b) Escrever a sub-rotina `prodMat` que calcula o produto resultante de  $A \times B$ , sendo A uma matriz  $N \times M$  e B uma matriz  $M \times P$ . Os parâmetros da sub-rotina são: *offset* do vetor que representa A, respetivo número de linhas e colunas, *offset* do vetor que representa B, respetivo número de colunas e *offset* do vetor produto ( $A \times B$  tem N linhas e P colunas). Note que cada elemento da matriz produto resulta do cálculo do produto interno de 2 vetores (vetor linha de A e vetor coluna de B).

**13.** Relativamente ao programa Viewer para tratamento de imagem, implemente sub-rotinas que executem as operações seguintes.

- a) Colocar a preto linhas alternadas da imagem.
- b) Reduzir para metade o valor de cada componente de cor da imagem.

Fim.