

Tratamento de dados de 8, 16 e 32 bits

1. Escrever um programa para determinar o comprimento de uma cadeia de caracteres terminada por zero e imprimir o resultado.
2. Implementar e testar programas que realizem as seguintes tarefas:
 - a) Copiar os dados de uma sequência de 10 bytes (com sinal) para uma sequência de 10 *words*.
 - b) Copiar os dados de uma sequência de 10 *words* (com sinal) para uma sequência de 10 *double words*.
 - c) Somar os elementos de uma sequência de 10 bytes (com sinal). O resultado deve ser representado no menor número de bits possível.
 - d) Repetir a alínea anterior para uma sequência de 10 *words*.
3. Escrever um programa para determinar se uma sequência de números inteiros do tipo `byte` está ordenado de forma crescente.
4. Escrever um programa que calcule a sequência que resulta da soma de elementos correspondentes de duas outras sequências de 10 palavras (16 bits, com sinal). Se a soma de duas palavras não for representável em 16 bits, usar o maior ou o menor valor representável, conforme essa soma seja, respetivamente, positiva ou negativa.
5. Por vezes, é necessário converter dados representados com N bits para representações com menos bits. Nesses casos, é preciso ter um critério para lidar com valores que estão fora da gama da nova representação.
 - a) Escrever um programa que copia os dados de uma sequência de 10 *double words* para uma sequência de *words*. Se o valor não for representável em 16 bits, usar o maior ou o menor valor representável, conforme se trate de um valor positivo ou negativo, respetivamente.
 - b) Repetir a alínea anterior para copiar uma sequência de *words* para uma sequência de bytes.
6. Escrever um programa para calcular a soma de todos os elementos de uma sequência de 1024 elementos do tipo `sbyte`. Guardar o resultado em EAX.

Instruções úteis

`movzx reg1, reg2/mem` realizar $\text{reg1} \leftarrow \text{reg2/mem}$, com `reg1` de tamanho superior a `reg2/mem`, extensão com 0.

`movsx reg1, reg2/mem` realizar $\text{reg1} \leftarrow \text{reg2/mem}$, com `reg1` de tamanho superior a `reg2/mem`, extensão de sinal.

Fim