

## Trabalho de Programação 1

### Processador RAMSES

#### 1. Descrição Geral

A empresa TABLETEROS S.A. armazena as **peças** usadas na fabricação de seus *ultrabooks* em **estantes** de um almoxarifado. Além disso, essas peças são classificadas em **conjuntos de montagem**. Por exemplo, um dos conjuntos de montagem é o “conjunto da CPU” (CPU, dissipador, parafusos, etc) ou o “conjunto da tela” (tela propriamente dita, *drivers* da tela, *leds* de iluminação, peças plásticas de sustentação, etc).

Nesse trabalho você deverá desenvolver um programa para o RAMSES capaz de calcular quantas PEÇAS de cada CONJUNTO existem em uma ESTANTE.

Para isso, você receberá uma lista das peças existentes na estante. Cada elemento dessa lista corresponde a uma determinada peça e contém duas informações: o CONJUNTO ao qual essa peça corresponde e a QUANTIDADE dessas peças existente na estante.

Para determinar o total de peças, de um determinado conjunto, existente na estante, basta somar o número de peças desse conjunto, avaliando cada um dos elementos da lista fornecida.

#### 2. Especificação da lista de entrada

Seu programa deverá ler as informações de FIM, CONJUNTO e QUANTIDADE dos elementos da lista fornecida e realizar o somatório adequadamente.

No endereço H80 da memória estará disponível o endereço do primeiro byte da lista (ou, como é comum se dizer, “no endereço H80 está o endereço da lista”). A lista propriamente dita estará armazenada em endereços consecutivos da memória, podendo iniciar em qualquer endereço a partir do H89.

A lista é formada por uma sequência de elementos. Cada elemento ocupa um byte. Cada byte tem o seguinte formato:

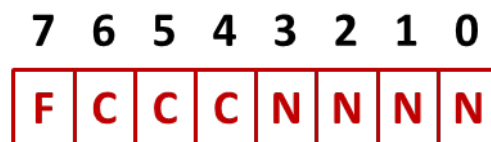


Figura 1 – Formato dos elementos da lista

Na figura 1 está representado um elemento da lista, que tem 1 byte de tamanho. Na figura estão representados os bits desse byte e uma letra de identificação. As letras significam o seguinte:

- **F:** Bits[7] – indicador de final da lista. Se for “1”, indica o final da lista. Nesse caso, os outros bits devem ser desconsiderados;
- **C:** Bits[6:4] – indica o número do conjunto ao qual pertence as peças representadas por esse elemento;
- **N:** Bits[3:0] – indica a quantidade de peças representadas por esse elemento.

### 3. Especificação do vetor de saída

Uma vez lidos os elementos de entrada (ou durante a leitura dos mesmos), seu programa deverá realizar a totalização das peças de cada conjunto de montagem. O resultado dessa soma deverá ser colocado em um vetor de totalizadores.

O vetor de totalizadores tem 8 elementos (um para cada conjunto de montagem). Além disso, cada elemento desse vetor tem 1 byte de tamanho e, como se trata de um totalizador, deverá ser usada uma representação de 8 bits sem sinal, conforme figura 2.

O vetor de totalizadores inicia no endereço H81. Assim, o elemento [0] está no endereço H81; o elemento [1] está no endereço H82 e assim sucessivamente até o elemento [7] que estará no endereço H88.

Vetor de totalizadores

H81	Conjunto de montagem 0
H82	Conjunto de montagem 1
H83	Conjunto de montagem 2
H84	Conjunto de montagem 3
H85	Conjunto de montagem 4
H86	Conjunto de montagem 5
H87	Conjunto de montagem 6
H88	Conjunto de montagem 7

Figura 2 – Vetor de totalizadores

OBS: Ao iniciar seu programa esse vetor deverá ser zerado.

### 4. Exemplo de cálculo

Suponha que tenha sido fornecida a seguinte lista de entrada:

H22, H43, H14, H65, H46, H27, H38, H39, H80

Separando cada um dos elementos da lista em seus componentes F, C e N, tem-se o seguinte:

- H22 → 0 010 0010 → F=0; C=2; N=2
- H43 → 0 100 0011 → F=0; C=4; N=3
- H14 → 0 001 0100 → F=0; C=1; N=4
- H65 → 0 110 0101 → F=0; C=6; N=5
- H46 → 0 100 0110 → F=0; C=4; N=6
- H27 → 0 010 0111 → F=0; C=2; N=7
- H38 → 0 011 1000 → F=0; C=3; N=8
- H39 → 0 011 1001 → F=0; C=3; N=9
- H80 → 1 000 0000 → F=1; Os valores presentes nos campos “C” e “N” não devem ser considerados.

O resultado da totalização deverá ser o seguinte:

- Totalizador[0]: 0                    MEM(H81) = H00
- Totalizador[1]: 4                    MEM(H82) = H04
- Totalizador[2]: 9                    MEM(H83) = H09
- Totalizador[3]: 17                   MEM(H84) = H11
- Totalizador[4]: 9                    MEM(H85) = H09
- Totalizador[5]: 0                    MEM(H86) = H00
- Totalizador[6]: 5                    MEM(H87) = H05
- Totalizador[7]: 0                    MEM(H88) = H00

## 5. Entregáveis: o que deve ser entregue?

---

Deverá ser entregue via Moodle da disciplina o arquivo fonte com a solução do problema apresentado, escrito em *assembly* do RAMSES (arquivo .RAD). Além disso, esse programa fonte deverá conter comentários descritivos da implementação. Sugere-se usar comandos da linguagem “C”.

Para a correção, o arquivo fonte do RAMSES será montado usando o montador Daedalus e serão aplicados 20 casos de teste. A nota final do trabalho será proporcional ao número de casos de teste em que o programa produzir a resposta correta (cada caso de teste vale 5,0 pontos de um total de 100,0).

O trabalho deverá ser entregue até a data prevista. **Não serão aceitos trabalhos entregues além do prazo estabelecido.**

Dentre os programas que fornecerem os resultados corretos para todos os casos de teste, aqueles mais rápidos (que utilizarem o menor número de acessos à memória) concorrerão a um “bônus” de 10,0 pontos na nota (10%).

## 6. Observações

---

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplinar Discente e a tomada das medidas cabíveis para essa situação (**tanto o trabalho original quanto os copiados receberão nota zero**).

O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.