

Trabalho de Programação 2

Processador CESAR

1. Descrição Geral

A empresa TABLETEROS S.A. assinou um contrato “milionário” de fornecimento de *ultrabooks* e teve que aumentar o número de estantes. Agora, a empresa armazena as **peças** usadas na fabricação de seus *ultrabooks* em várias **estantes** de um almoxarifado. Da mesma forma que no programa para o RAMSES, as peças são classificadas em **conjuntos de montagem**.

Nesse trabalho você deverá desenvolver um programa para o CESAR capaz de calcular, a partir das informações das PEÇAS de cada CONJUNTO existentes nas ESTANTES, os seguintes valores:

- Número de estantes existentes no almoxarifado;
- Número total de peças de cada conjunto existente no almoxarifado;
- Número de peças de cada conjunto existente em uma determinada estante.

Para isso, você terá acesso a listas: uma LISTA DE ESTANTES, que representa o almoxarifado, e várias LISTAS DE PEÇAS, cada uma representando uma estante. Portanto, cada elemento da LISTA DE ESTANTES aponta para uma LISTA DE PEÇAS existentes naquela estante. As informações relativas a essas listas poderão ocupar os endereços de memória que iniciam em 32768 (8000_{16}) até 36863 ($8FFF_{16}$) e serão inseridas na memória no momento da correção do trabalho. Portanto, seu programa não deve escrever nessa área nem alocar espaço para variáveis na mesma. Caso o espaço da primeira metade da memória (endereços 0 até 32767) não seja suficiente para o programa, pode-se reservar os endereços a partir de 36864 (9000_{16}) usando ORG 32864 ou ORG H9000. Entretanto, deve-se tomar cuidado com a área de periféricos e a pilha.

Cada elemento da LISTA DE PEÇAS corresponde a uma determinada peça, e contém duas informações: o CONJUNTO ao qual essa peça pertence e a QUANTIDADE dessas peças existente na estante.

Para calcular a quantidade de peças de um determinado conjunto, existente na estante ou no almoxarifado, basta somar o número de peças desse conjunto, avaliando cada um dos elementos da lista fornecida.

2. Especificação da LISTA DE ESTANTES – almoxarifado

O endereço da LISTA DE ESTANTES estará armazenado no endereço 32768 (8000_{16}) da memória. Os elementos dessa lista estarão dispostos de forma contígua na memória, como é comum a vetores de elementos.

Cada elemento da LISTA DE ESTANTES tem 16 bits e contém um endereço absoluto, que indica o primeiro elemento de uma LISTA DE PEÇAS. Esse modelo de dados está representado na Figura 1.

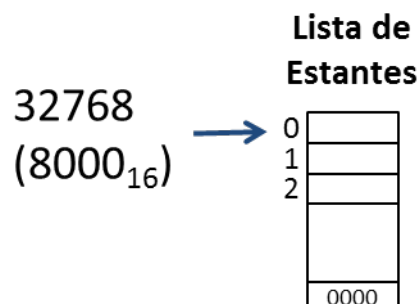


Figura 1 – Lista de Estantes - almoxarifado

Ainda, conforme pode ser visto na Figura 1, o final da LISTA DE ESTANTES é indicado por uma palavra contendo o valor zero (“NULL”).

3. Especificação da LISTA DE PEÇAS – estante

Os elementos da LISTA DE PEÇAS estão dispostos de forma contígua na memória. Cada elemento contém as informações de FIM, CONJUNTO e QUANTIDADE e ocupa uma palavra de 16 bits, conforme o formato apresentado na Figura 2.

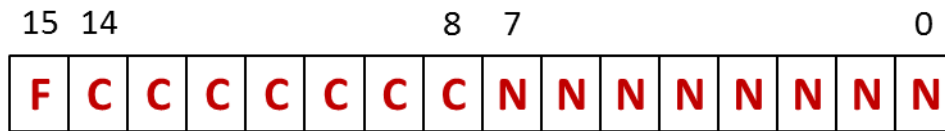


Figura 2 – Elemento da LISTA DE PEÇAS

Na Figura 2 está representado um elemento da lista, que tem dois bytes (uma WORD) de tamanho. Na figura, os bits dessa WORD estão identificados por letras, que significam:

- **F:** Bits 15 – indicador de final da lista. Se for “1”, indica o final da lista. Nesse caso, os outros bits devem ser desconsiderados;
- **C:** Bits 14 a 8 – indica o número do conjunto ao qual pertencem as peças representadas por esse elemento (0 a 127);
- **N:** Bits 7 a 0 – indica a quantidade de peças representadas por esse elemento (0 a 255).

4. Resumo da “área de dados”

Na Figura 3 está representada a estrutura de dados que deverá ser usada pelo programa. O endereço da LISTA DE ESTANTES estará sempre armazenada no endereço 32768 (8000_{16}) da memória. Nessa lista, cada elemento contém o endereço de memória no qual inicia uma LISTA DE PEÇAS armazenadas em uma estante (ponteiro para a estante). O final da lista de estantes é indicado por um elemento com o valor zero.

Cada LISTA DE PEÇAS corresponde a uma estante do almoxarifado e será formada por uma ou mais palavras com o formato descrito na Figura 2.

O final da lista de peças é indicado por um elemento com o bit 15 igual a 1. Os outros campos desse elemento final devem ser ignorados.

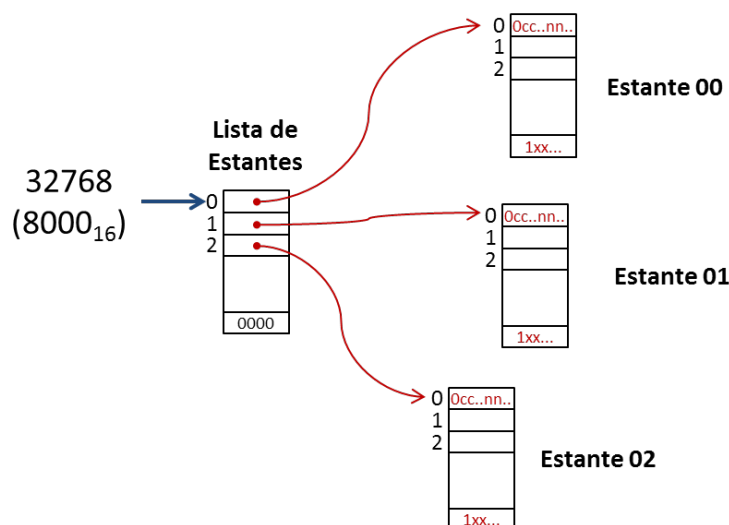


Figura 3 – Diagrama da Área de Dados

5. Especificação dos procedimentos a serem implementados

O programa deverá operar conforme o diagrama apresentado nas Figura 4 e Figura 5. Seus blocos componentes estão descritos logo após as figuras.

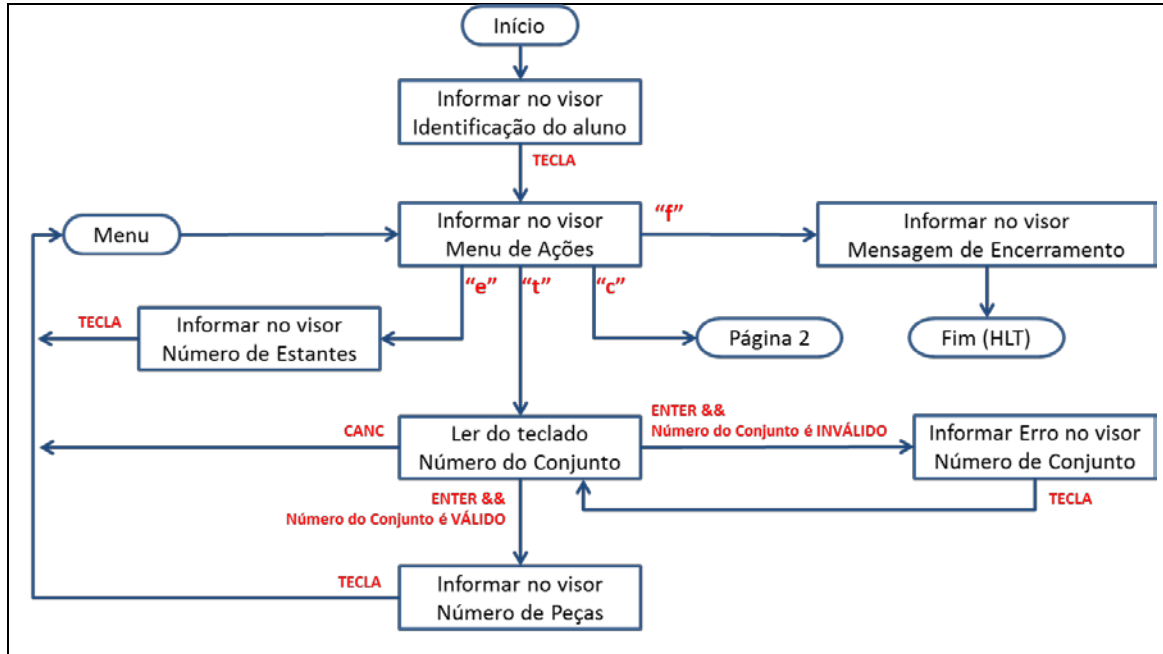


Figura 4 - Algoritmo a ser implementado – página 1

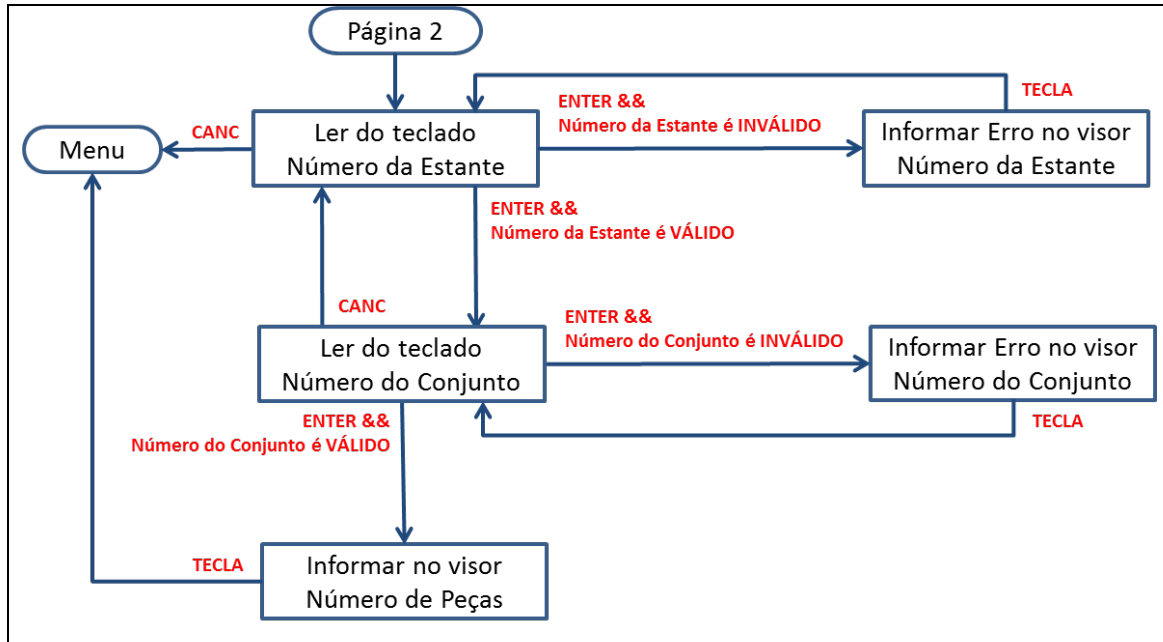


Figura 5 - Algoritmo a ser implementado – página 2

Informar no visor – Identificação do aluno: O programa deve colocar no visor o nome e o número de cartão do aluno. Depois disso, o programa fica aguardando que seja digitada qualquer tecla (TECLA).

Informar no visor – Menu de Ações: O programa deve apresentar no visor o pedido que o usuário selecione uma das ações do programa. Após colocar essa mensagem, o programa deve aguardar que seja digitada uma tecla com a opção selecionada. O programa deve verificar a validade da tecla e, então, executar a ação correspondente (conforme diagrama da Figura 5). Teclas inválidas devem ser ignoradas e o programa deve continuar nesse mesmo bloco de

operação. Nenhuma das teclas (válidas ou inválidas) deve ser exibida no visor. As teclas válidas são: “e” (número total de estantes no almoxarifado), “t” (número total de peças de um conjunto no almoxarifado), “c” (número total de peças de um conjunto, existente em uma determinada estante) e “f” (fim).

Informar no visor – Número de Estantes: O programa deve determinar o número de estantes e apresentar esse valor no visor. Esse número terá, no máximo, 3 dígitos. Zeros à esquerda devem ser suprimidos.

Informar no visor – Número de Peças: O programa deve calcular o número de peças de certo conjunto. No caso desse bloco da Figura 5, esse número será o totalizador de peças do almoxarifado; no caso desse bloco na Figura 5, esse número será o totalizador de peças da estante informada. Em ambos os casos esse número poderá ter até 4 dígitos. Zeros à esquerda devem ser suprimidos.

Ler do teclado – Número da Estante: O programa deve ler um número decimal inteiro sem sinal do teclado, conforme descrito no item 6: “Procedimento para ler um número do teclado”. Quando nesse bloco, pode ocorrer uma de três situações:

- Ser digitada a tecla CANCEL (ASCII 27 ou 1B₁₆);
- Ser digitada a tecla ENTER (ASCII 13 ou 0D₁₆) e o valor numérico ser INVÁLIDO;
- Ser digitada a tecla ENTER e o valor numérico ser VÁLIDO.

A cada uma dessas situações corresponde uma ação, descrita nos diagramas das Figura 4 e Figura 5. Se o valor numérico digitado for válido, então esse será considerado o “Número da Estante” para efeito nos próximos blocos de processamento.

Ler do teclado – Número do Conjunto: Comportamento semelhante ao anterior, exceto que, se o valor numérico digitado for válido, esse será considerado o “Número do Conjunto”, para efeito nos próximos blocos de processamento.

Informar Erro no visor – Número da Estante: Informa que foi introduzido um Número da Estante inválido.

Informar Erro no visor – Número do Conjunto: Informa que foi introduzido um Número do Conjunto inválido.

6. Procedimento para ler um número do teclado

Na especificação do trabalho resumida nas Figura 4 e Figura 5 existem dois blocos de procedimento que envolve a leitura de um número do teclado. Esse número pode ser o “Número da Estante” ou o “Número do Conjunto”.

Para a entrada desses números o programa deverá tratar as teclas válidas: teclas dos dígitos entre “0” e “9”, a tecla ENTER, a tecla BS – *back space* – (ASCII 08 ou 08₁₆) e a tecla CANCEL (Cancelar). Outras teclas devem ser ignoradas.

- As teclas de dígitos devem ser colocadas no visor, movendo o cursor para a direita, até um máximo de dígitos que se pretende ler. Após esse limite, essas teclas também passam a ser ignoradas;
- A tecla BS deve recuar o cursor para a esquerda, apagando o dígito anterior. Cuidado especial deve ser tomado quando o cursor estiver na posição mais a esquerda, pois não há dígitos a serem apagados;
- A tecla ENTER indica que o usuário encerrou a entrada do número e deverá ser validado;
- A tecla CANCEL indica que o usuário desistiu de entrar o número e deseja que eventuais digitações anteriores sejam ignoradas. Essa tecla também encerra a entrada do número.

7. 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 CESAR (arquivo .CED). 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 CESAR será montado usando o montador Daedalus e serão aplicados casos de teste. A nota final do trabalho será proporcional às funcionalidades que forem atendidas pelo programa.

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

8. 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.