

Trabalho de Programação 1 - Processador RAMSES

1. Descrição Geral

O objetivo deste trabalho é desenvolver para o processador Ramses um programa capaz de calcular o número de ladrilhos em uma parede. Para isso, uma foto da parede foi obtida e processada por um programa de reconhecimento de imagens capaz de identificar ladrilhos bons e quebrados. Como resultado desse processamento, é gerado um vetor de bytes que será fornecido para o seu programa contar o total de ladrilhos da parede, os ladrilhos quebrados e os ladrilhos em bom estado.

2. Dados do vetor de bytes de entrada

O vetor de bytes com as informações dos ladrilhos da parede estará na memória. O endereço de início do vetor estará armazenado no endereço 208 ($D0_{16}$) e o vetor estará armazenado em algum lugar na área delimitada pelas posições de memória 213 ($D5_{16}$) e 255 (FF_{16}).

Cada byte do vetor possui informação referente a até 4 ladrilhos, que estão lado a lado na parede, sendo utilizados dois bits para codificar o estado de um ladrilho.

Na Figura 1 está representada uma parede com ladrilhos (quadrados). Cada 4 ladrilhos, correspondem a um byte (indicado por um retângulo vermelho). Na Figura 1 estão indicados os ladrilhos que correspondem aos bytes 0 e 2 do vetor.

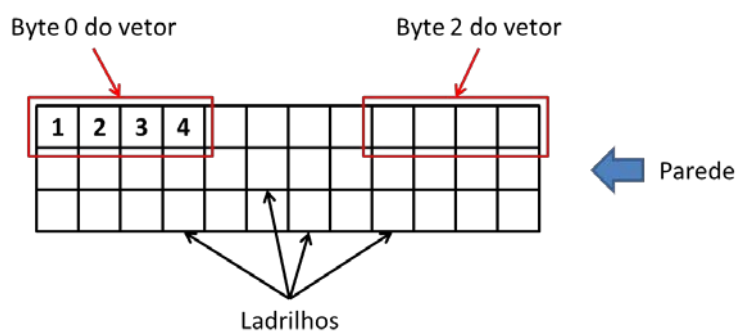


Figura 1

A Figura 2 mostra como a informação de quatro ladrilhos está armazenada em bytes na memória. Conforme indicado na Figura 2, o ladrilho 1 (representado como o ladrilho mais à esquerda na parede na Figura 1) está representado nos bits 7 e 6; o ladrilho 2 nos bits 5 e 4, e assim por diante, até o ladrilho 4.

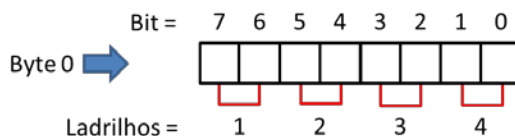


Figura 2

Para codificar o estado de cada ladrilho, usando dois bits, será utilizada a seguinte convenção:

- Bits = 00, indica que não existe ladrilho, por se tratar de uma porta ou janela;
- Bits = 01, indica que o ladrilho está em boas condições;
- Bits = 10, indica que o ladrilho está danificado e deve ser substituído;
- Bits = 11, indica o fim do vetor de ladrilhos.

3. Quantidades a calcular

Com base no vetor de bytes com as informações dos ladrilhos, calcule as seguintes informações:

- Número de ladrilhos **danificados** (Código 10₂). Esse contador deve ser escrito na posição de memória 209 ($D1_{16}$);

- Número de ladrilhos em **boas condições** (Código 01₂). Esse contador deve ser escrito na posição de memória 210 (D2₁₆);
- Número de ladrilhos **não utilizados**, correspondentes às portas e janelas (Código 00₂). Esse contador deve ser escrito na posição de memória 211 (D3₁₆);
- Número **total de ladrilhos** usados para cobrir a parede, conforme informados no vetor de bytes. Esse contador deve ser escrito na posição de memória 212 (D4₁₆).

4. Processamento

O processamento a ser realizado pelo programa consiste em percorrer todos os elementos do vetor, identificando a situação de cada um dos quatro ladrilhos codificados nos seus bytes e incrementando o contador apropriado. O final do vetor é identificado pelo código "11₂" na posição de um ladrilho. É garantido que sempre exista essa indicação de final do vetor e que os códigos no vetor estão todos corretos.

Exemplo: supondo que os dados de entrada para o programa sejam os seguintes:

Endereço	208 (D0 ₁₆)	...	224 (E0 ₁₆)	225 (E1 ₁₆)	226 (E2 ₁₆)
Dado	224 (E0 ₁₆)		6 (06 ₁₆)	6 (06 ₁₆)	87 (57 ₁₆)

Separando os dados do vetor de bytes (06₁₆, 06₁₆ e 57₁₆) em seus ladrilhos correspondentes, obtêm-se o seguinte vetor de ladrilhos (valores em binário):

00 00 01 10 . 00 00 01 10 . 01 01 01 11

Que correspondem a ladrilhos nos seguintes estados:

Vazio Vazio Bom Danificado Vazio Vazio Bom Danificado Bom Bom Bom Fim

O resultado do processamento para os valores contidos no vetor será, portanto, o seguinte:

Endereço do contador	209 (D1 ₁₆)	210 (D2 ₁₆)	211 (D3 ₁₆)	212 (D4 ₁₆)
Conteúdo (decimal)	2	5	4	7

Notar que:

- Se necessário, os dados originalmente fornecidos no vetor de bytes podem ser alterados pelo programa;
- Na correção serão verificados apenas os quatro bytes correspondentes aos contadores.

5. Correção dos Trabalhos

Os arquivos fonte do RAMSES entregues serão montados usando o montador DAEDALUS. A seguir serão aplicados 20 casos de teste, de forma sequencial e contínua (sem recarga ou inicialização da memória, apenas alterando o endereço do vetor, o conteúdo do vetor e apontando o PC para 0). 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 pontos de um total de 100).

6. Bônus

Os programas que fornecerem resultados corretos para todos os 20 casos de teste poderão concorrer a um Bônus de Desempenho, no valor de 10 pontos (10% da nota). Esse bônus será concedido para as soluções que utilizarem o menor número de **acessos** à memória (aqueles mais rápidos). O desempenho será medido somando os números de **acessos** usados para solucionar todos os 20 casos de teste.

7. Entregáveis: o que deve ser entregue?

Somente os arquivos fonte (arquivos .RAD) escritos na linguagem simbólica do RAMSES com a solução do problema apresentado deverão ser entregues via Moodle da disciplina. **Esses arquivos serão montados com o DAEDALUS.** O programa fonte deverá conter comentários descritivos da implementação. Por exemplo, nos comentários podem ser

usados comandos da linguagem “C”. Sugere-se usar a primeira letra de seu nome seguida de 6 dígitos do número de cartão para dar nome ao arquivo .RAD.

O trabalho deverá ser entregue até a data especificada no link de entrega no sistema Moodle. **Não serão aceitos trabalhos após o 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.

9. Casos de Teste

Para testar sua solução, use os casos de teste abaixo, da seguinte forma:

1. Carregue seu código (arquivo .MEM gerado pelo Daedalus a partir do seu .RAD) no simulador Ramses
2. Coloque os valores indicados nas palavras 208 e 213 a 255 (as palavras em branco ou não mostradas na tabela não precisam ser preenchidas)
3. Coloque o valor 0 no PC
4. Execute o programa
5. Verifique se os resultados nas palavras 209 a 212 são os mesmos indicados no final da linha de cada caso de teste
6. Para testar os demais casos, volte à etapa 2 **SEM RECARREGAR O CÓDIGO DE SUA SOLUÇÃO NO SIMULADOR**

Caso	End. →	208	213	214	215	216	217	218	219	220	221	222	223	224	225	226	...	252	253	254	255	209	210	211	212
1	Dados	224												6	6	87	...					2	5	4	7
2		214		192													...					0	0	0	0
3		220								24	192						...					1	1	2	2
4		215			24	3											...					1	1	5	2
5		218						85	85	192							...					0	8	0	8
6		255															...				3	0	0	3	0
7		252															...	170	170	170	171	15	0	0	15
8		213	170	85	170	85	170	85	192								...					12	12	0	24
9		222										1	2	3			...					1	1	9	2
10		217					0	0	0	0	0	0	0	0	0	3	...					0	0	39	0