

## Trabalho de Programação – Processador RAMSES

### 1. Descrição Geral

Nesse trabalho você deverá desenvolver um programa para o processador RAMSES que seja capaz de separar as informações contidas em um vetor de números que está na memória em dois outros vetores derivados e, ao final dessa separação, efetuar o somatório dos elementos indicados de um desses vetores derivados.

O vetor original é formado por bytes armazenados na memória a partir do endereço HF1 ( $241_{10}$ ). O final de vetor original está marcado por um valor HFF ( $255_{10}$ ). Os valores armazenados nesse vetor podem ser **números** ou **índices**. Os números são todos os valores entre H00 ( $0_{10}$ ) e H7F ( $127_{10}$ ). Os índices são todos os valores entre H80 ( $128_{10}$ ) e HFE ( $254_{10}$ ). Observar que a diferenciação entre números e índices é feita pelo conteúdo do bit 7 do valor.

Seu programa deverá separar o vetor original em dois outros vetores derivados. Um desses vetores deverá conter apenas números, e o outro vetor apenas índices. Ao fazer essa separação, seu programa também deverá zerar o bit 7 dos índices, de tal forma que todos os índices armazenados no vetor derivado de índices serão valores entre H00 ( $0_{10}$ ) e H7F ( $127_{10}$ ).

Depois de feita a separação, seu programa deverá verificar a consistência dos valores do vetor de índice. Isso significa verificar se os índices são valores válidos para indicar os elementos no vetor de números. Caso seja encontrado um índice que não tenha elemento correspondente no vetor de números, esse índice deverá ser considerado inválido e não deverá ser utilizado no restante do processamento. Observar que o primeiro elemento do vetor de números é aquele identificado pelo índice “0” (zero).

Finalmente, seu programa deverá efetuar a soma de todos os elementos do vetor de números cujos índices forem válidos e estiverem no vetor de índices. Deve ser realizada uma soma de valores de 8 bits e o eventual *overflow* deve ser ignorado.

Seu programa e as variáveis necessárias ao processamento deverão ser colocados na memória a partir do endereço H00 ( $0_{10}$ ) e não deverão ultrapassar o endereço HEF ( $239_{10}$ ).

### 2. Disposição dos dados na memória

#### Informações de entrada

O endereço do primeiro elemento do vetor original é HF1 ( $241_{10}$ ). O vetor encerra quando for encontrada a marca de final de vetor (HFF). É garantida a existência da marca de final do vetor.

#### Informações de saída

O resultado final do processamento (somatório dos números indicados pelos índices válidos) deverá ser escrito no endereço HF0 ( $240_{10}$ ).

### 3. Restrições e garantias

- Seu programa e variáveis de trabalho devem estar restritos a região de memória H00 até HEF;
- É garantido que o vetor original possua a marca de final de vetor;
- Quaisquer dos vetores (original os derivados) podem ser vazios. Em qualquer desses casos, o programa deve informar “0” (zero) como resultado do somatório;
- 

### 4. Exemplo

Suponha que na memória, no espaço reservado para o vetor original, estejam disponíveis os seguintes valores (representados em hexa-decimal):

Endereço	F0	F1	F2	F3	F4	F5	F6	F7
Dado	01	82	02	81	03	04	8F	FF

Depois da primeira etapa de separação do vetor original nos dois vetores derivados, obtêm-se os seguintes dois vetores.

Vetor de números

01	02	03	04
----	----	----	----

Vetor de índices

02	01	0F
----	----	----

Observa-se que, como resultado dessa etapa de processamento, o vetor de números contém 4 (quatro) elementos. Observa-se, também, que os valores identificados como índices tiveram os seus respectivos bits 7 zerados e que, portanto, contém os índices "02" e "01", que são válidos, e o índice "0F" ( $15_{10}$ ), que é inválido.

Como o índice inválido não deve ser usado, vamos removê-lo do vetor. Ficam os vetores, então, da seguinte forma:

Vetor de números

01	02	03	04
----	----	----	----

Vetor de índices

02	01
----	----

Finalmente, procede-se o somatório dos elementos indicados pelo vetor de índice. Deve-se efetuar a soma dos elementos "02" e "01" do vetor de números (indicado por "NUM", para compactar a expressão).

Somatório = NUM[02] + NUM[01]

Somatório = 03 + 02

Somatório = 05

O resultado final ("05") deve ser escrito no endereço HF0 ( $240_{10}$ ) da memória.

## 5. Correção dos Trabalhos

---

Os arquivos fonte do RAMSES entregues serão montados usando o montador DAEDALUS. A seguir serão aplicados 10 (dez) casos de teste, de forma sequencial e contínua, sem recarga ou inicialização da memória. O único ajuste que será realizado entre um teste e outro é o preenchimento do vetor original. A nota final do trabalho será proporcional ao número de casos de teste em que os programas produzirem a resposta correta.

## 6. O que deve ser entregue?

---

Deverão ser entregues somente os arquivos fonte (arquivos .RAD) escritos na linguagem simbólica do RAMSES, com a solução do problema apresentado, no 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".

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

## 7. 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.