

Exercícios

Exercício 1 (*level easy*)

(soma dos elementos de uma matriz)

Escrever um programa para o computador RAMSES capaz de preencher uma matriz com o resultado da soma dos seus índices de linha e coluna. A matriz é quadrada com dimensões $N \times N$ bytes ($N \in [2;8]$). A dimensão da matriz (valor de N) está no endereço H80 e o elemento (1,1) da matriz deverá estar no endereço H90. Considere que a numeração de linha e coluna inicia em “1”

Por exemplo, o elemento (4,5) deverá ser preenchido com “9” ($4+5=9$)

Exercício 2 (*level medium*)

(soma de matrizes)

Escrever um programa para o computador RAMSES para realizar a soma de duas matrizes quadradas de dimensões 4 x 4 bytes. As matrizes a serem somadas são a matriz “A” e a matriz “B^T” (a transposta de “B”).

A matriz “A” inicia no endereço H80, a matriz “B” inicia no endereço H90 e a matriz “S” (resultado da soma) inicia no endereço “HA0”.

Para calcular a soma de cada elemento pode-se utilizar a seguinte expressão:

$$s_{ij} = a_{ij} + b_{ji}$$

Exercício 3 (*level medium*)

(soma da diagonal principal de uma matriz)

Escrever um **programa** para o computador RAMSES capaz de calcular a soma de todos os elementos da diagonal principal de uma matriz quadrada de ordem 5. Todos os elementos da matriz assim como a soma são valores de oito bits sem sinal.

As variáveis estão declaradas em “C” da seguinte forma:

```
unsigned char Matriz[5][5];
```

```
unsigned char Soma;
```

E estão declaradas em *assembly* do NEANDER da seguinte forma:

```
ORG    H80
```

```
Soma:  db      0
```

```
Matriz: dab    [25]
```

Você deve implementar, em *assembly*, o programa a seguir:

```
Soma = 0;
```

```
for (i=0; i<5; ++i)
```

```
    Soma = Soma + Matriz[i][i];
```

Exercício 4 (*level hard*)

(matriz transposta)

Escreva um programa para o processador RAMSES para realizar o cálculo da matriz transposta de ordem $N \times N$, onde N pode ser um número entre 2 e 7. Seu programa e dados de cálculo podem ocupar o espaço entre os endereços 0 (H00) e 127 (H7F). O restante da memória é ocupado da seguinte forma:

- A ordem da matriz está definida no endereço 128 (H80);
- O primeiro elemento da matriz original ("M") está no endereço 130 (H82);
- O primeiro elemento da matriz transposta ("T") calculada está no endereço 180 (HB4).

Obs1: os dados das matrizes estão armazenadas em linhas. Ou seja, os elementos da matriz estão armazenados, endereço por endereço, da seguinte forma: $M(0,0)$, $M(0,1)$, $M(0,2)$, ..., $M(N-1,N-1)$. Observe que estão sendo usados valores entre "0" e "N-1" para os índices da matriz.

Obs2: um possível algoritmo para o cálculo da matriz transposta é o seguinte:

```
for (i=0; i<N; i++)  
    for (j=0; j<N; j++)  
        T( j , i ) = M( i , j );
```

Exercício 5 (*level hard*)

(série de Fibonacci)

Escreva um programa para o processador RAMSES capaz de gerar, na memória, a Série de Fibonacci, a partir de dois números consecutivos da série fornecidos. Seu programa e dados de cálculo podem ocupar o espaço entre os endereços 0 (H00) e 127 (H7F). O restante da memória é ocupado da seguinte forma:

- Número de termos a serem gerados (além daqueles já fornecidos), no endereço 128 (H80)
- Número inicial da série, no endereço 129 (H81)
- Próximo número da série, no endereço 130 (H82)
- Números gerados da série, a partir do endereço 131 (H83)

Obs1: É garantido que os dois números fornecidos são consecutivos e pertencem à série de Fibonacci.

Obs2: Os cálculos devem ser feitos com 8 bits. Caso o resultado do cálculo seja maior do que 255 (HFF), este deverá ser truncado nos 8 bits menos significativos. Esse valor deverá ser colocado na memória e usado nos próximos cálculos.