

Atividade de Laboratório 3

[Atividade de Laboratório 3](#)

[Objetivos](#)

[Descrição](#)

[Requisitos \(ATENÇÃO!\)](#)

[Dicas](#)

[Avaliação](#)

[Entrega](#)

Objetivos

O objetivo desta atividade é exercitar o conceito de laço e o método de acesso indireto à memória do computador IAS.

Descrição

Escreva um programa em linguagem de máquina IAS (usando mapa de memória) para calcular o produto escalar de dois vetores. Produto escalar é uma operação que realiza a soma dos produtos de todos os elementos de 2 vetores, dois a dois. Por exemplo, para os vetores $A=(1,2)$ e $B=(4,13)$, temos o seguinte produto escalar: $1*4+2*13 = 30$. Você pode supor que os vetores possuem o mesmo tamanho, contudo esse tamanho é arbitrário. Os vetores podem ou não estar representados consecutivamente no mapa de memória. Mais informações são apresentadas na seção abaixo.

Requisitos (ATENÇÃO!)

- O tamanho dos vetores deve ser lido da memória, no endereço 0x3FD;
- Os vetores possuem pelo menos um elemento;
- Os endereços dos vetores (endereços do primeiro elemento de cada vetor) devem ser lidos da memória (endereços 0x3FF e 0x3FE);
- Seu mapa de memória deve ser organizado em 3 partes:
 - Na primeira parte há o código (instruções) e dados;
 - Na segunda parte (o "meio" do mapa de memória) estão os 2 vetores;
 - Nas 3 últimas posições estão armazenados os endereços iniciais e o tamanho dos vetores, conforme explicado no item acima.
- Observe que, para submissão, somente a primeira parte será necessária;
- Mantenha o código e as variáveis que você criar no início da memória. Em especial, não escreva código e suas variáveis a partir do endereço 0x100, pois os *scripts* de teste utilizados para validação do seu programa podem armazenar os vetores de teste nestas posições.
- Termine a execução com um salto para o endereço 0x400;

- Antes de terminar a execução, salve o resultado do produto escalar no registrador AC. A corretude será verificada através da inspeção do valor neste registrador.

Dicas

- O arquivo (blank.hex) contém **todas** as posições de memória do IAS preenchidas com 0 - pode ser útil como "molde".
- Para facilitar o projeto e a implementação do programa, você pode considerar que a multiplicação gera apenas resultados de até 40 *bits*.

Avaliação

Para avaliar seu programa, executaremos o mesmo com 10 entradas diferentes e verificaremos o resultado armazenado no registrador AC.

A avaliação será realizada com o mecanismo de automação de testes do simulador IAS, disponível na página:

<http://www.ic.unicamp.br/~edson/disciplinas/mc404/2017-2s/abef/IAS-sim/testmodule.html>.

Veja uma explicação sobre este mecanismo na atividade de [laboratório 2](#).

O vetor de testes a seguir define os 5 casos de testes públicos desta atividade de laboratório.

```
[ // five open test cases
{
  "input": [
    {"where": "ram", "position": "0x3FD", "value": 3}, // size of vectors
    {"where": "ram", "position": "0x3FF", "value": "0x211"}, // address of v1
    {"where": "ram", "position": "0x3FE", "value": "0x222"}, // address of v2

    {"where": "ram", "position": "0x211", "value": 0},
    {"where": "ram", "position": "0x212", "value": 2},
    {"where": "ram", "position": "0x213", "value": 5},

    {"where": "ram", "position": "0x222", "value": 1},
    {"where": "ram", "position": "0x223", "value": 6},
    {"where": "ram", "position": "0x224", "value": 9}

  ],
  "output": [{"where": "reg", "position": "ac", "value": 57}]
},
{
  
```

```

"input": [
  {"where": "ram", "position": "0x3FD", "value": 1}, // size of vectors
  {"where": "ram", "position": "0x3FF", "value": "0x200"}, // address of v1
  {"where": "ram", "position": "0x3FE", "value": "0x201"}, // address of v2

  {"where": "ram", "position": "0x200", "value": 101},

  {"where": "ram", "position": "0x201", "value": 4}
],
"output": [{"where": "reg", "position": "ac", "value": 404}]
},

{
  "input": [
    {"where": "ram", "position": "0x3FD", "value": 5}, // size of vectors
    {"where": "ram", "position": "0x3FF", "value": "0x200"}, // address of v1
    {"where": "ram", "position": "0x3FE", "value": "0x280"}, // address of v2

    {"where": "ram", "position": "0x200", "value": 0},
    {"where": "ram", "position": "0x201", "value": 0},
    {"where": "ram", "position": "0x202", "value": -5587},
    {"where": "ram", "position": "0x203", "value": 0},
    {"where": "ram", "position": "0x204", "value": 0},

    {"where": "ram", "position": "0x280", "value": -10},
    {"where": "ram", "position": "0x281", "value": 15},
    {"where": "ram", "position": "0x282", "value": 0},
    {"where": "ram", "position": "0x283", "value": 435333480},
    {"where": "ram", "position": "0x284", "value": -558870}

  ],
  "output": [{"where": "reg", "position": "ac", "value": 0}]
},

{
  "input": [
    {"where": "ram", "position": "0x3FD", "value": 5}, // size of vectors
    {"where": "ram", "position": "0x3FF", "value": "0x200"}, // address of v1
    {"where": "ram", "position": "0x3FE", "value": "0x300"}, // address of v2

    {"where": "ram", "position": "0x200", "value": 6},
    {"where": "ram", "position": "0x201", "value": 0},
    {"where": "ram", "position": "0x202", "value": 128},

```

```

        {"where": "ram", "position": "0x203", "value": -432},
        {"where": "ram", "position": "0x204", "value": 1},

        {"where": "ram", "position": "0x300", "value": 63},
        {"where": "ram", "position": "0x301", "value": -12},
        {"where": "ram", "position": "0x302", "value": -404},
        {"where": "ram", "position": "0x303", "value": 612},
        {"where": "ram", "position": "0x304", "value": 0}

    ],
    "output": [{"where": "reg", "position": "ac", "value": -315718}]
},

{
    "input": [
        {"where": "ram", "position": "0x3FD", "value": 10}, // size of vectors
        {"where": "ram", "position": "0x3FF", "value": "0x300"}, // address of v1
        {"where": "ram", "position": "0x3FE", "value": "0x380"}, // address of v2

        {"where": "ram", "position": "0x300", "value": 334},
        {"where": "ram", "position": "0x301", "value": 22},
        {"where": "ram", "position": "0x302", "value": -982},
        {"where": "ram", "position": "0x303", "value": -10},
        {"where": "ram", "position": "0x304", "value": -919},
        {"where": "ram", "position": "0x305", "value": 911},
        {"where": "ram", "position": "0x306", "value": 7000},
        {"where": "ram", "position": "0x307", "value": -1},
        {"where": "ram", "position": "0x308", "value": 182},
        {"where": "ram", "position": "0x309", "value": 9},

        {"where": "ram", "position": "0x380", "value": 324},
        {"where": "ram", "position": "0x381", "value": 237},
        {"where": "ram", "position": "0x382", "value": 423},
        {"where": "ram", "position": "0x383", "value": 345},
        {"where": "ram", "position": "0x384", "value": 8},
        {"where": "ram", "position": "0x385", "value": 43},
        {"where": "ram", "position": "0x386", "value": 1},
        {"where": "ram", "position": "0x387", "value": 288},
        {"where": "ram", "position": "0x388", "value": 177},
        {"where": "ram", "position": "0x389", "value": 0}

    ],
    "output": [{"where": "reg", "position": "ac", "value": -234659}]
}

```

}
]

Para saber se sua solução está passando por todos os testes, inclusive os fechados, utilize esse link: <https://cloud.lmcad.ic.unicamp.br/test>

Entrega

- Você deve submeter APENAS um arquivo no Moodle, chamado raXXXXXX.hex, em que XXXXXX é seu ra com 6 dígitos.