

Instruções:

1. Esta atividade pode ser realizada individualmente ou em **até TRÊS** alunos (no máximo).
2. O relatório deverá ser postado no Material Didático, com uma capa identificando a Instituição, o curso, a disciplina, o professor, o nome da atividade, os autores do trabalho e data em que o mesmo for entregue.
3. A defesa da implementação (vídeo de no máximo 5 minutos) e postagem do relatório **15/11/2023** as 23:59. Defesa e/ou entrega em atraso **não serão aceitas**.
4. Os arquivos da implementação devem ser compactados em um único arquivo **ZIP** e postados no ambiente Material Didático, acompanhados do relatório em formato **PDF**. Antes de compactar, organize o seu modelo em uma pasta denominada: **Cache_Nome1_Nome2**, onde Nome(1 a 2) refere-se aos nomes dos integrantes do grupo. Alternativamente poderá ser enviado um link para o github contendo o código e o vídeo.
5. Se forem identificados trabalhos com grau de similaridade que caracterize cópia ou adaptação (autorizadas ou não pelos seus autores originais), a nota dos grupos será a nota de um trabalho dividida pelo número de grupos que entregou esses trabalhos similares.

ATIVIDADE

Enunciado:

Implemente um programa que realize a soma de duas matrizes quadradas e armazene o resultado em uma terceira matriz, com opção de percorrer as matrizes em linha-coluna e coluna-linha, e analise as taxas de acertos das diferentes possibilidades de organização da cache de dados.

Requisitos:

1. Na seção de declaração de variáveis (`.data`), cada matriz deve ser declarada com um espaço para 100 elementos. Elas devem ser claramente identificadas com nomes como `Matriz_A`, `Matriz_B` e `Matriz_C`, por exemplo.
2. O programa deve solicitar o número de linhas/colunas das matrizes aceitando no máximo 10 elementos. Para leitura, deve ser apresentada uma mensagem solicitando a entrada desse valor, indicando o seu limite máximo. Ex: "Entre com o tamanho do índice das matrizes (máx. = 10).".

- O programa deve solicitar a entrada do número de elementos até que ele seja **maior que 1 e menor ou igual a 10**. Ou seja, deve implementar um mecanismo de filtragem que não aceite entrada diferente da especificada. No caso de entrada inválida, o programa deve imprimir uma mensagem de advertência antes de solicitar novamente a entrada. Ex: "Valor inválido".
- O conteúdo das matrizes, antes da execução da função de soma, deve corresponder à seguinte estrutura:

```

Matriz_A = 0 0 0 0 0 0 0 ... 0
Matriz_B = 1 1 1 1 1 1 1 ... 1
Matriz_C = 0 1 2 3 4 5 6 ... 99

```
- O conteúdo das matrizes pode ser declarado no segmento de dados (.data) ou então preenchido por meio de código **antes da execução das operações de soma**.
- O programa deve solicitar a forma com que as matrizes serão percorridas, sendo '0' para linha-coluna e '1' para coluna-linha. No caso de entrada inválida, o programa deve imprimir uma mensagem de advertência antes de solicitar novamente a entrada. Ex: "Valor inválido".
- Os programas devem ser escritos respeitando o estilo de programação ASM, usando tabulação para organizar o código em colunas (rótulos, mnemônicos, operandos e comentários).
- Procure comentar ao máximo o seu código. Isto é um hábito da programação *assembly*.

Análise:

Após a implementação, analise a taxa de acertos de **TODAS** as combinações de cache possíveis para uma cache de **1024 bytes** (Cache size (bytes) = 1024) na ferramenta "Data Cache Simulator" do RARS.

OBS: mude apenas os parâmetros destacados em vermelho que resultem em um cache size de 1024.

The screenshot shows the 'Cache Organization' window with the following settings:

- Placement Policy: Direct Mapping
- Block Replacement Policy: LRU
- Set size (blocks): 1
- Number of blocks: 16
- Cache block size (words): 16
- Cache size (bytes): 1024

A red box highlights the 'Number of blocks' and 'Cache block size (words)' fields, which are set to 16 and 16 respectively.

Para a análise, considerar matrizes com 10 linhas/colunas e variar o acesso em linha-coluna e coluna-linha. Apresentar a taxa de acerto em **TODAS** as combinações.

No relatório, indicar as organizações de cache com a melhor e a pior taxa de acertos para cada forma de acesso e justificar porque acredita que a organização deu esses resultados