

## Trabalho de Programação – Processador INTEL – 2025/1

### Descrição Geral

Desenvolva um programa na linguagem de montagem do INTEL X86 capaz de ler dois arquivos:

- um arquivo de dados, que contém uma matriz de números;
- um arquivo de expressões, que contém várias expressões aritméticas.

Seu programa deverá ser capaz de realizar as expressões listadas no arquivo de expressões, usando os números que estão no arquivo de dados.

No final, seu programa deverá escrever os resultados obtidos das expressões, em um arquivo de resultados.

**Obs:** todos os números que estão no arquivo de dados deverão ser representados em complemento de 2 com 16 bits. Se houver overflow na operação, o resultado deverá ser truncado em 16 bits.

### Formato do arquivo DADOS

Este arquivo terá, sempre, o nome “DADOS.TXT”, e terá a seguinte estrutura:

- A primeira linha terá o número de colunas da matriz (N). Este valor deverá estar entre 1 e 20 (valores diferentes devem ser identificados como erro);
- As outras linhas terão “N” números separados por “;” (ponto e vírgula).

É garantido que:

- Não haverá linhas vazias (exceto, possivelmente, a última linha);
- Não haverá espaços (SPACE, TAB, etc) entre os números de uma linha;
- Os números dispostos nas linhas serão separados por um caractere “;” (ponto e vírgula). Cada um destes números corresponde a uma coluna.

Identifica-se o final das linhas ao encontrar o final do arquivo ou ao encontrar uma linha vazia.

Entretanto, pode ocorrer que uma linha esteja com o número errado de colunas. Seu programa deve identificar este erro e terminar o programa caso isso aconteça.

Os números desta matriz estão representados como números inteiros com sinal. Deve ser possível representá-los sem overflow, em complemento de 2 com 16 bits.

Exemplo de arquivo DADOS.TXT

```
3
10;123;-15
2;-3;54
```

Este arquivo indica que estão representados nas linhas a seguir informações para 3 (três) colunas.

No arquivo existem duas linhas, formadas por números positivos e negativos, separados por “;”, sem qualquer espaço entre eles.

### Formato do arquivo EXP

Este arquivo terá, sempre, o nome “EXP.TXT”, e conterá as expressões aritméticas a serem realizadas sobre o arquivo de dados.

Cada linha do arquivo corresponde a uma expressão a ser calculada. Assim, identifica-se o final das expressões ao encontrar o final do arquivo ou ao encontrar uma linha vazia. Pode haver até 100 expressões.

## Formato das expressões

As expressões de uma linha do arquivo EXP são escritas no formato **[linha\_resultado]=[operação e operandos]**. O campo *linha\_resultado* representa a linha que deve ser atualizada com o resultado da operação, considerando 0 (zero) como índice da primeira linha. Os operandos podem ser constantes ou referências para outras linhas.

Por exemplo, para o campo *linha\_resultado*:

- [10] está **correto** (desde que o arquivo de entrada tenha 11 ou mais linhas)
- \*[8] está **correto**, sendo que o caractere "\*" representa escrita no arquivo de resultados (descrição a seguir);
- [-7] está **incorreto**, pois não há indexação de linhas negativas.

As expressões são seguidas do símbolo "=" e, em seguida, da expressão propriamente dita. É garantido que não haverá espaços entre os elementos que formam a expressão.

Para construir as expressões, podem ser usadas constantes ou referências para linhas entre colchetes.

- **Constantes:** poderão ser representadas em complemento de 2 com 16 bits;
- **Referências entre colchetes:** indicam uma linha da matriz de dados. Caso seja usado uma referência errada, seu programa deve identificar e informar o erro.

Assim, são exemplos de expressões.

- [0]=[0]+[2]      Soma dos valores das linhas [0] e [2] da matriz e escreve resultado na linha 0
- \*[1]=5\*[0]      Multiplicação da linha [0] da matriz por 5, escreve o resultado na linha 1
- [10]=[2]+15      Soma 15 à linha [2] e escreve o resultado na linha 10

## Operações nas expressões

As expressões devem ser capazes de realizar as seguintes operações, que devem ser implementadas pelas instruções do processador:

Símbolo	Descrição
+	Soma
-	Diferença
*	Multiplicação com sinal
/	Divisão com sinal (16 bits / 16 bits, com resultado com 16 bits)
%	Resto (16 bits / 16 bits, com resultado com 16 bits)
&	Lógica AND
	Lógica OR
^	Lógica XOR (ou exclusivo)

Não há parênteses nas expressões. Caso necessário, para realizar alguma operação, o usuário deve utilizar expressões intermediárias.

Caso seja utilizada uma operação inválida (diferente da tabela anterior), seu programa deve identificar e informar o erro.

## Geração dos resultados

---

Os resultados, conforme descritos a seguir, devem ser escritos em um arquivo de nome "RESULT.TXT".

Devem ser escritos no arquivo de resultados apenas os resultados das expressões que tiverem o símbolo "\*" junto ao seu identificador no arquivo "EXP.TXT". No exemplo abaixo, apenas os resultados dos cálculos da expressão  $*[1]=...$  deverão ser colocados no arquivo de resultados.

- $[0]=...$
- $*[1]=...$
- $[2]=...$

Os resultados serão organizados em grupos de linhas, onde cada grupo representa uma operação. Cada grupo de linhas deve ser descrito da seguinte forma

- A primeira linha do grupo descreve a operação (ex:  $*[0]=[0]+[2]$ )
- As N linhas consecutivas representam a matriz resultante após a operação.

## Correção dos Trabalhos

---

Para corrigir os trabalhos serão aplicados casos de teste, formados por arquivos DADOS.TXT e EXP.TXT e, em seguida, serão verificados os resultados na tela (identificação de situações de erro) e o conteúdo do arquivo RESULT.TXT.

Os arquivos fonte entregues serão montados usando o MASM no ambiente DOSBOX. Programas que possuam erros de montagem receberão nota 0 (zero).

O código do programa fonte deverá conter comentários descritivos da implementação.

O trabalho deverá ser entregue até a data prevista indicada no sistema Moodle. Não serão aceitos trabalhos entregues após o prazo estabelecido. Os trabalhos não entregues até a data prevista receberão nota zero.

## O que deve ser entregue?

---

Deverá ser entregue somente o arquivo fonte (arquivo ASM) escrito na linguagem simbólica do INTEL X86, com a solução do problema apresentado, no Moodle da disciplina. **Não entregue o arquivo EXE.**

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.**

## 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. Inicialmente, nesses casos, **ambos os trabalhos: original e cópias, 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.