



Universidade Federal da Bahia (UFBA)  
Instituto de Matemática e Estatística (IME)  
Departamento de Ciências da Computação (DCC)  
MATA49 – T01 – Programação de Software Básico  
Prof. Igo Amaurí dos Santos Luz

## Primeiro Trabalho Prático

### Objetivo:

Consolidar aprendizagem acerca da programação em assembly para Intel IA-32 (x86) através de um problema matemático.

### Descrição do Trabalho:

Nas expressões matemáticas escritas na notação infixa os operadores são escritos entre os operandos. Uma operação com dois operandos, seguindo a notação infixa, deve ser escrita seguindo uma determinada ordem: o primeiro operando, o símbolo da operação e, por fim, o segundo operando. Nessa notação torna-se necessário utilizar regras de precedência, ou prioridade, entre as operações, para isso, utilizam-se parênteses.

Em um grupo de no máximo 3 pessoas, os alunos da disciplina MATA49 – T01 devem desenvolver um programa em assembly para a arquitetura Intel IA-32 que tenha como entrada uma expressão matemática na notação infixa e realize as seguintes manipulações:

- Verificar se a expressão se encontra bem formatada em relação aos parênteses;
- Caso esteja bem formatada, apresentar o resultado da expressão.

O programa deve ser capaz de efetuar operações em ponto flutuante.

### Entrada:

O programa será testado através da inserção de uma expressão no terminal.

Exemplos de expressões:

```
2 + 3 * 4  
2 * ( 3 + 4 ) / 7 - 1  
5 * ( 8 + 9 ) * ( 7 - 3 ) * 2  
( ( 6 + 1 ) - 5 * 4  
( 9 + 5 ) / 3
```

**Saída:**

O programa deve apresentar o resultado para cada expressão no terminal. Considerando o exemplo de entrada descrito acima, a saída apresentada para cada expressão deverá ser:

**Expressão 1:**

Bem formatada

14

**Expressão 2:**

Bem formatada

1

**Expressão 3:**

Bem formatada

680

**Expressão 4:**

Erro de formatação

**Expressão 5:**

Bem formatada

4,66

**Relatório**

Cada equipe deverá escrever um relatório descrevendo o desenvolvimento do trabalho. O relatório deverá abordar, pelos menos, os seguintes tópicos:

- Breve introdução
- Descrição do algoritmo implementado.
- Decisões de Implementação.
- Conclusão.
- Bibliografia utilizada

**Informações Importantes:**

- O trabalho pode ser desenvolvido em equipe de no máximo 3 pessoas.
- O trabalho deve ser entregue no Moodle até o dia **05/07/2018**. As apresentações irão ocorrer no dia **11/07/2018**.
- Somente um integrante da equipe deverá submeter o arquivo compactado contendo o código fonte e o relatório no Moodle.
- Qualquer identificação de plágio, o trabalho será penalizado com zero.
- O trabalho deverá ser desenvolvido utilizando o assembler NASM e no ambiente Linux.

- Os arquivos fontes e o relatório deverão ser entregues na tarefa no Moodle em um mesmo arquivo compactado. Somente um integrante do grupo precisa submeter o arquivo no Moodle.

### **Apresentação:**

Cada equipe terá, no máximo, 15 minutos para apresentar o funcionamento do seu trabalho.

É obrigatória a presença de todos os integrantes da equipe na apresentação. Em caso de não comparecimento, o integrante da equipe será penalizado com a pontuação referente ao entendimento individual sobre a solução implementada.

A avaliação do entendimento individual poderá ser realizada através da apresentação de cada integrante e/ou arguições feitas pelo professor.

### **Cronograma**

06/06/2018 (Quarta-Feira) – Início do Trabalho

11/06/2018 (Segunda-Feira) – Aula Teórica

13/06/2018 (Quarta-Feira) – Discussão do Trabalho

18/06/2018 (Segunda-Feira) – Aula Teórica – Avaliação Teórica

20/06/2018 (Quarta-Feira) – Aula de Laboratório

23/06/2018 (Sábado) – Discussão do Trabalho

25/06/2018 (Segunda-Feira) – Aula Teórica

05/07/2018 (Quinta-Feira) – Entrega do trabalho

11/07/2018 (Quarta-Feira) – Apresentação do trabalho

**Barema de Avaliação:**

Grupo		
Integrantes:		
CrITÉrios de Avaliação	Pontuação Máxima	Pontuação Atribuída
Código devidamente comentado	1,0	
Organização do código	1,5	
Implementação e funcionamento da solução	5,5	
Relatório	2,0	
Entendimento individual sobre a solução implementada	- 4,0	
Nota Final:		
Observações:		