

# PROJETO DE DISCIPLINA INTRODUÇÃO À PROGRAMAÇÃO

## 1a Verificação de Aprendizagem

1. Você deverá codificar, usando a Linguagem C, um simulador assembly;
2. O seu programa deverá, ao final de cada instrução digitada, imprimir o conteúdo dos registradores de propósito geral e da memória de dados;
3. Seu programa deverá simular as seguintes instruções:
  - a. **MOV** *registrador, constante*
  - b. **MOV** *endereço de memória, constante*
  - c. **MOV** *endereço de memória, registrador*
  - d. **MOV** *registrador, endereço de memória*
  - e. **ADD** *registrador, registrador*
  - f. **ADD** *registrador, constante*
  - g. **HLT** *fim da simulação*
4. Você deve implementar a mesma sintaxe (e arquitetura) do simulador visto em sala de aula, isso quer dizer que:
  - a. a sintaxe [45] refere-se ao conteúdo armazenado no endereço de memória 45;
  - b. o primeiro parâmetro da instrução **MOV** é o destino e o segundo a fonte do dado;
  - c. Os registradores de propósito geral são nomeados como **A, B, C e D**;

## Exemplo de execução do simulador (entrada de dados em negrito)

```
** Bem vindo ao Simulador Assembly em Linguagem C **
Banco de registradores: A:0 B:0 C:0 D:0
Memória de dados: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MOV B,25
Banco de registradores: A:0 B:25 C:0 D:0
Memória de dados: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MOV A,31
Banco de registradores: A:31 B:25 C:0 D:0
Memória de dados: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ADD A,B
Banco de registradores: A:56 B:25 C:0 D:0
Memória de dados: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
MOV [8],A
Banco de registradores: A:56 B:25 C:0 D:0
Memória de dados: 0 0 0 0 0 0 0 0 56 0 0 0 0 0 0 0 0 0 0 0 0 0
HLT
```

### **Dicas e sugestões**

1. Use a função `gets()` para ler uma string que pode conter espaços;
2. Considere que cada string de instrução é composta por: 3 caracteres de instrução + espaço (ascii 32) + primeiro parâmetro + vírgula (ascii 44) + segundo parâmetro;
3. Leia os caracteres do parâmetro procurando por caracteres '[' e ']' os quais indicam referência à memória;
4. Cheque se algum dos parâmetros refere-se às letras que indicam registradores;
5. Quando necessário, converta uma string em número natural usando a função `atoi()`;