Trabalho 1

- **Desenvolvimento de programa em linguagem de montagem MIPS**
- **Ferramenta usada**: **MARS**
  - MIPS Assembler and Runtime Simulator
  - http://courses.missouristate.edu/KenVollmar/MARS/
  - Instalada nos laboratórios
- **Objetivos**:
  - Familiarização com:
    - Conjunto de instruções MIPS
    - Técnicas de programação em linguagem de montagem
- **Programa**: **Ordenação de vetor e visualização**
  - **Bubble Sort** (método da bolha)
    - Algoritmo de ordenação simples
  - **Exemplo**:
    - www.hackerearth.com/pt-br/practice/algorithms/sorting/bubble-sort/visualize/

Programa Principal: Ordenação

- **Algoritmo**:

Entrada de dados: armazenados na memória
- n: número de elementos do vetor
- vetor de n inteiros

```
Imprime mensagem inicial
mostra_vetor()
trocou = true
limite = n - 1
while (limite > 0) AND (trocou)
    trocou = false
    for (i = 0 ; i < limite ; i++)
        if vetor[i] > vetor[i+1]
            troca vetor[i] e vetor[i+1] na memória
            trocou = true
    mostra_vetor()
    limite--
Encerra execução do programa
```

Rotina mostra_vetor()

- Mostra no display bitmap cor correspondente a todos os elementos de vetor
- **Possui chamada aninhada de rotina**:
  - Chama rotina mostra_elemento_vetor
- **Algoritmo**:

```
Salva endereço de retorno ($ra) na pilha
Salva registradores na pilha (se necessário)
for (j = 0 ; j < n ; j++)
    mostra_elemento_vetor(j)
Restaura endereço de retorno ($ra) da pilha
Restaura registradores da pilha (se necessário)
```

Programa

- **Implementar**:
  - Programa principal (com ordenação)
  - Rotina mostra_vetor
- **Rotina fornecida pronta**:
  - mostra_elemento_vetor

Importante

- **Inicialmente**:
  - Entender programa BubbleSort.asm fornecido (incompleto)
  - Entender algoritmo de ordenação pelo método da bolha
- **Em seguida: desenvolver o programa completo**
  - Planejar alocação de variáveis na memória
  - Planejar uso dos registradores
  - Traduzir algoritmo para linguagem de montagem:
    - Manter lógica das estruturas de controle: laços, if, ...
  - **Para rotina**:
    - Planejar passagem de parâmetros por registradores
    - Planejar salvamento/restauração de endereço de retorno e registradores na pilha
- **Colocar comentários**:
  - Uso dos registradores, trechos de código
- **Não modificar manipulação do display**
- **Para executar**:
  - **Configurar memória e display de acordo com orientações fornecidas**
  - Testar programa para diferentes vetores
- **Simplificação**: n = 16, elementos do vetor com valores entre 0 e 15

Ideias de Melhorias e Extensões: Opcional

- **Entrada de dados**:
  - Ler valor de n do teclado
  - Ler elementos do vetor do teclado ou de um arquivo de entrada
- **Saída de dados**:
  - Escrever elementos do vetor **ordenado** na tela ou em um arquivo de saída
- **Implementar outros algoritmos de ordenação**:
  - Insertion Sort
  - Selection Sort
  - Merge Sort
  - ...
- **Eliminar simplificações**:
  - Elementos do vetor poder ter quaisquer valores
- **Outras formas de visualização**
- ...

Entrega do Trabalho

- **Grupos**: 2 alunos
- **Programa fornecido**: BubbleSort.asm (incompleto)
- **Entrega do trabalho**:
  - **Submissão no Moodle**
  - **Submeter um único arquivo BubbleSort.asm** com:
    - Programa desenvolvido em linguagem de montagem MIPS
    - Nome dos alunos do grupo em comentários
- **Data de entrega**: