

# Trabalho Primeiro EP

Prof. Dr. Clodoaldo A. de Moraes Lima

Material baseado no livro "Patterson, David A., Hennessy, J. L.  
- Computer Organization And Design: The Hardware/Software  
Interface"



Atualmente, o mundo se vê as voltas com problemas para tratamento de uma quantidade de dados cada vez maior. Apesar dos computadores atuais atingirem um desempenho muitas vezes superior aos do passado, a quantidade de informações gerada e processada também aumentou exponencialmente.

O desafio de hoje é manter os dados indexados para que possam ser referenciados e buscados com um menor esforço. Esta maneira de armazenar as informações requer muitas vezes que os dados sejam ordenados segundo um determinado critério.

Neste trabalho deve ser implementado dois métodos de ordenação de dados. Um com complexidade  $O(n^2)$  e outro com  $O(n \log n)$

## Exemplo

Dado o seguinte vetor de dados em valores numéricos armazenado em um arquivo:

3 2 6 8 9 10 4 7 12 15

A solução seria os seguintes valores armazenados em arquivo:

2 3 4 6 7 8 9 10 12 15

## O que deve ser feito

- Sua tarefa neste trabalho será implementar dois algoritmo na linguagem assembly que resolva o problema acima.
- Considere que o conjunto inicial de informações (vetor inicial) esteja armazenado em um arquivo.
- O tamanho máximo do vetor é denotado por  $n$ , não é conhecido a priori.
- O Aluno deve varer o arquivo para identificar o numero de elementos

# Algoritmos a serem implementados

- Cálculo do valor de N
  - Some o Número USP de todos os membros do trabalho
  - Divida a soma do Número USP pelo número de membros do trabalho
  - Atribua a N o dígito mais à direita do Número USP médio
  - Exemplo - Número USP médio = 8082623, N será igual a 3
- Se N estiver no intervalo  $[0, 2.5]$ , os algoritmos a serem implementados são o método de inserção e o Quicksort.
- Se N estiver no intervalo  $]2.5, 5]$ , os algoritmos a serem implementados são o método de seleção e o Quicksort.
- Se N estiver no intervalo  $]5, 7.5]$ , os algoritmos a serem implementados são o método de bolha e o Quicksort
- Se N estiver no intervalo  $]7.5, 10]$ , os algoritmos a serem implementados são o método de eliminação e o Quicksort

# Requisitos

A função principal deve chamar uma função de ordenação que deve obrigatoriamente possuir a seguinte assinatura:

## Requisito 1

`int ordena(int tam, int tipo, int *vetor)`, onde:  
retorno (`int`): Tamanho do vetor resultante (este valor muda no caso do método de eliminação);  
`ordena`: Nome da função (também em assembly);  
`int tam`: Tamanho do vetor a ser ordenado;  
`int tipo`: especifica qual método a ser utilizado  
`int *vetor`: Vetor com os valores a serem ordenados

## Requisito 2

Os dados devem ser lidos de um arquivo, o qual será passado como entrada para o programa. O vetor ordenado de forma crescente deve ser escrito no final do arquivo.

## O que deve ser entregue?

- Documentação descrevendo a lógica principal da sua implementação;
- Código do fonte do seu programa em assembly;
- Documentação e código-fonte devem ser enviados como anexo para o Tidia

## Quem? Quando?

- Grupos de 4 alunos.
- Data da Entrega até o dia 08/11/2020 (23:59).

- Utilize o simulador MARS para executar e testar seu programa. Nele é possível acompanhar passo-a-passo a execução do algoritmo, além de saber o valor de cada um dos registradores.  
(<http://courses.missouristate.edu/kenvollmar/mars/>). Listagem das chamadas de sistema disponíveis no MARS: <http://courses.missouristate.edu/kenvollmar/mars/help/syscallhelp.html>.
- De modo a facilitar seu trabalho, codifique primeiro o programa em linguagem de alto-nível e depois simplesmente o traduza para assembly MIPS.
- Utilize como referência os trechos de código abaixo. Eles podem ser úteis na leitura dos dados digitados pelo usuário no MARS.

```
.data
str_num: .asciiz    "Quantos    numeros deseja  inserir?(maximo 200)\n"
.text
.globl main
main:
        la    $a0, str_num
        li    $v0, 4
        syscall
        li    $v0, 5
        syscall
```

PATTERSON, D. A., HENNESSY, J. L., Organização e Projeto de Computadores: A Interface Hardware/ Software, Ed. Campus, 5a ed., 2014.