

**Departamento de Informática - UFES**  
**Estruturas de Dados: 2º Trabalho Prático**  
**Período: 2019/2**  
**Profª Patrícia Dockhorn Costa**  
**Email: edufes2019@gmail.com**

*Data de Entrega: 04/12/2019*

*Grupos de 2 pessoas*

***Regras Importantes***

- Não é tolerado plágio. Trabalhos copiados serão penalizados com zero.
- **A data de entrega é inadiável.** Não serão tolerados atrasos neste trabalho.

***Material a entregar***

- Documentação do trabalho, que deve conter:
  - Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  - Implementação: descrição da implementação do programa. Devem ser detalhadas as estruturas de dados utilizadas (de preferência com diagramas ilustrativos), o funcionamento das principais funções utilizadas incluindo pré e pós condições, o formato de entrada e saída de dados, bem como decisões tomadas relativas aos casos e detalhes de especificação que porventura estejam omissos no enunciado. Modularize o seu programa como discutido em sala de aula (usando TAD's).
  - Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  - Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso.
- Por email (não é necessário relatório impresso):
  - O assunto da mensagem deve ser ed2019:trab2:<nome1>:<nome2>
    - Por exemplo: ed2019:trab2:<joaosilva>:<mariacosta>
  - Documentação do trabalho (em formato PDF).
  - Todos os arquivos .c e .h criados (e xigido código muito bem documentado!).
  - O makefile.

## Compactador/Descompactador de Arquivos

Pede-se neste trabalho para implementar um compactador/descompactador de arquivos baseado na codificação de Huffman, como descrito no material disponibilizado pela professora.

O seu programa *Compacta* deverá ler os dados de entrada a partir de um arquivo, cujo nome é passado como parâmetro na linha de comando (Faz parte do trabalho descobrir como manipular arquivos em C). O programa *Compacta* deverá gerar um arquivo que representa o arquivo compactado (*nome\_arquivo.comp*). Exemplo de execução do programa a partir da linha de comando:

```
Compacta nome_arquivo.txt
```

O arquivo de entrada pode ser qualquer arquivo binário.

De maneira similar, o seu programa *Descompacta* deverá ler os dados de entrada a partir de um arquivo, cujo nome é passado como parâmetro na linha de comando. O programa *descompacta* deve gerar o arquivo original (*nome\_arquivo.txt*). Exemplo de execução do programa a partir da linha de comando:

```
Descompacta nome_arquivo.comp
```

**BOM TRABALHO!**