

# Atividade Prática 01

## Conteúdo: Pilhas

Universidade Tecnológica Federal do Paraná (UTFPR), campus Apucarana  
Curso de Engenharia de Computação  
Disciplina de Estrutura de Dados - ED62A - 2º Semestre 2019  
Prof. Dr. Rafael Gomes Mantovani

### 1 Descrição

Elabore um programa em C que realize a conversão de números decimais para números binários usando pilhas estáticas e/ou dinâmicas. Use as implementações das estruturas desenvolvidas em sala para resolver o problema. O programa receberá dois arquivos texto como parâmetros de entrada:

- **arquivo de entrada:** um arquivo texto contendo os números decimais a serem convertidos. A primeira linha do arquivo contém um caractere único, especificando qual tipo de pilha será usada: 'd' - pilha dinâmica, ou 'e' - pilha estática. As demais linhas que existirem no arquivo conterão números decimais (um por linha), que deverão ser convertidos;
- **arquivo de saída:** um arquivo texto onde serão impressos os correspondentes números binários, porém na ordem inversa a qual foram lidos. Um número binário por linha.

Um exemplo de arquivo de entrada e saída é apresentado na Figura 1.

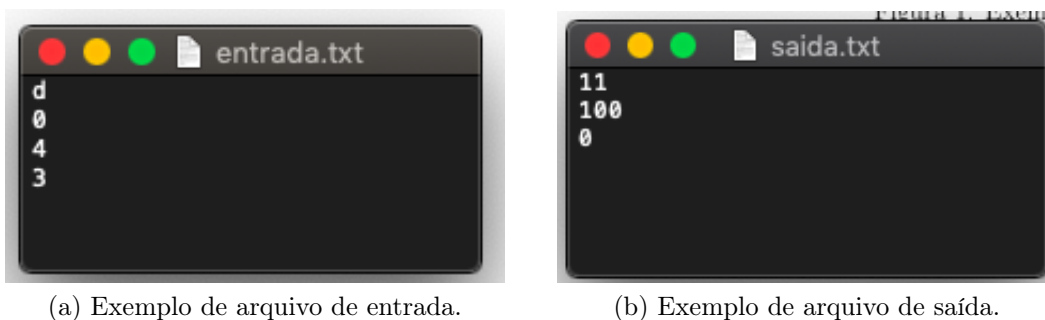


Figura 1: Valores de entrada e correspondente saída do programa. Atente para o fato de que a impressão dos valores correspondentes é contrária à ordem de leitura.

**Dica:** Quando for rodar o programa por linha de comando (run), é necessário manipular os argumentos **argc** e **argv** da função **main**. Para isso, deve-se executar o programa por linha de comando, obedecendo o seguinte padrão:

<nome do programa> <arquivo de entrada> <arquivo de saída>

Por exemplo, se o programa fonte se chamar “conversor.c”, o comando será:

**conversor entrada.txt saida.txt**

Dentro da função `main`, o valor de `argc` indica o número de parâmetros recebidos para a execução. Como são inseridos apenas os nomes dos arquivos de entrada e saída, `argc = 3` (os dois arquivos mais o nome do programa). Além disso, `argv[0]` contém o nome do programa, `argv[1]` o nome do arquivo de entrada, e `argv[2]` o nome do arquivo de saída.

## 2 Orientações gerais

- Implementar também o controle de erros, para lidar com exceções e erros que possam ocorrer;
- Para acompanhamento do desenvolvimento, criar um repositório individual com o código desenvolvido no github Classroom, por meio do link: <https://classroom.github.com/a/818iHEPc>. Os repositórios serão privados, com acesso apenas do professor e do aluno.
- Entrega do programa final: via Moodle. O aluno deve submeter o fonte (apenas os arquivos .c) no link da atividade disponibilizado na página da disciplina no Moodle.
- **Data de entrega:** 16/09/2019.
- Os códigos desenvolvidos por cada aluno serão também verificados por ferramentas de plágio. Códigos iguais/similares terão nota zero.

## 3 Links úteis

Arquivos em C:

- <https://www.inf.pucrs.br/~pinho/LaproI/Arquivos/Arquivos.htm>
- <https://www.geeksforgeeks.org/basics-file-handling-c/>
- <https://www.programiz.com/c-programming/c-file-input-output>

Argumentos de Linha de comando (`argc` e `argv`):

- [https://www.tutorialspoint.com/cprogramming/c\\_command\\_line\\_arguments.htm](https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm)
- <http://linguagemc.com.br/argumentos-em-linha-de-comando/>
- [http://www.univasf.edu.br/~marcelo.linder/arquivos\\_pc/aulas/aula19.pdf](http://www.univasf.edu.br/~marcelo.linder/arquivos_pc/aulas/aula19.pdf)
- [http://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-31\\_Argumentos\\_linha\\_comando.html](http://www.inf.ufpr.br/cursos/ci067/Docs/NotasAula/notas-31_Argumentos_linha_comando.html)
- <http://www.dca.fee.unicamp.br/cursos/EA876/apostila/HTML/node145.html>