

# Atividade Prática 01

## Manipulação de Pilhas

---

Universidade Tecnológica Federal do Paraná (UTFPR), campus Apucarana  
Curso de Engenharia de Computação  
Disciplina de Estrutura de Dados - EDCO3A  
Prof. Dr. Rafael Gomes Mantovani

---

### Instruções:

- Leia todas as instruções corretamente para poder desenvolver sua atividade/programa;
- Evite plágio (será verificado por meio de ferramentas automatizadas). Faça seu programa com os seus nomes de variáveis e lógica de solução. Plágios identificados anularão as atividades entregues de todos os envolvidos.
- Adicione comentários nos códigos explicando seu raciocínio e sua tomada de decisão. Porém, não exagere nos comentários, pois a própria estrutura do programa deve ser auto-explicativa.
- Salve sua atividade em um arquivo único, com todas as funções e procedimentos desenvolvidos. É esse **arquivo único** que deverá ser enviado ao professor.

## 1 Descrição da atividade

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. Na atividade, iremos manipular a informação por meio de arquivos texto. Assim, 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 arquivos de entrada e saída válidos é apresentado na Figura 1. Percebamos que o arquivo de entrada (na esquerda) especifica o uso de pilhas dinâmicas, indicado pelo caractere 'd' na primeira linha; seguido de três números decimais que deverão ser convertidos (0, 4, 3). No arquivo de saída (na direita), temos a impressão dos correspondentes números binários, porém em ordem inversa à entrada, isto é, (3, 4, 0) em binário.

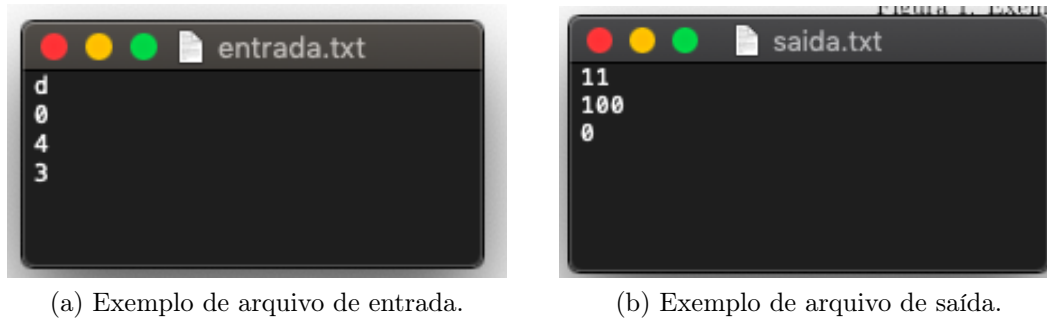


Figura 1: Valores de entrada e correspondente saída do programa. Atentem para o fato de que a impressão dos valores correspondentes é contrária à ordem de leitura.

**Dica:** Quando forem executar o programa com os arquivos texto 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 “conversorD2B.c”, o comando será:

**conversorD2B 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, nesse caso temos **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

Além da funcionalidade desejada, implementar também o controle de erros, para lidar com exceções que possam ocorrer, como por exemplo:

- problemas nas aberturas dos arquivos de entrada e saída;
- arquivo de entrada vazio (sem informação);

- arquivo de entrada fora do padrão esperado (opções inválidas para tipo da pilha, ou números que não sejam inteiros nas demais linhas);
- etc.

Opcionalmente, para acompanhamento do desenvolvimento, pode-se criar um repositório individual no `github`.

## 2.1 Critério de correção

A nota na atividade será contabilizada levando-se em consideração alguns critério:

1. pontualidade na entrega;
2. não existir plágio;
3. completude da implementação (tudo foi feito);
4. o código compila e executa;
5. uso de `argc` e `argv` para controle dos arquivos de teste;
6. implementar o parser para entrada dos dados via arquivo texto;
7. implementação das duas estruturas necessárias (estática/dinâmica);
8. legibilidade do código (identação, comentários nos blocos mais críticos);
9. implementação dos controles de erros (arquivos de entrada inválidos, e erros no programa principal);
10. controle de memória: chamar o destrutor e desalocar a memória de tudo se usar a estrutura dinâmica, fechar os arquivos, etc;
11. executar corretamente os casos de teste.

Em cada um desses critérios, haverá uma nota intermediária valorada por meio de conceitos: **Sim** - se a implementação entregue cumprir o que se esperava daquele critério; **Parcial** - se satisfizer parcialmente o tópico; e **Não** se o critério não foi atendido.

## 2.2 Dados para envio da atividade

Ao elaborar seu programa, crie um único arquivo fonte (.c) seguindo o padrão de nome especificado:

`ED1-<ANO>-<SEMESTRE>-AT01-ConvertorD2B-<NOME>.c`

Exemplo:

`ED-2021-2-AT01-ConvertorD2B-RafaelMantovani.c`

A entrega da atividade será via Moodle: o link será disponibilizado na página da disciplina.

### 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>

### Referências

- [1] Thomas H. Cormen,; Ronald Rivest; Charles E. Leiserson; Clifford Stein. Algoritmos - Teoria e Prática - 3ª Ed. Elsevier - Campus, 2012.
- [2] Nivio Ziviani. Projeto de algoritmos com implementações: em Pascal e C. Pioneira, 1999.
- [3] Adam Drozdek. Estrutura De Dados E Algoritmos Em C++. Cengage, 2010.