

Estrutura de Dados 2

Laboratório 4 – Simulado da Prova 1

1 Instruções

Siga as instruções a seguir para essa atividade de laboratório.

- Procure resolver todas as questões abaixo sem realizar nenhuma consulta (à Internet, a anotações ou livros, ou ao colega) para simular o cenário de prova.
- Veja o código C incompleto fornecido para cada questão. Complete-o para resolver o exercício. Você é livre para criar novas funções, macros e arquivos `.c` ou `.h`. Apenas certifique-se de editar o **Makefile** adequadamente de forma que um simples comando **make** seja suficiente para compilar o executável da questão.
- Todas as questões possuem uma entrada e saída fornecida e devem ser executadas e testadas com o seguinte comando:

```
$ ./exi < in | diff out -
```

onde `exi` é o executável do exercício.

2 Questões

1. Crie um programa que implementa uma calculadora *pós-fixada* com as operações de soma e multiplicação. O seu programa deve ler a expressão a ser calculada de `stdin` e imprimir o resultado em `stdout`. A expressão de entrada nunca tem mais de 50 operandos. Utilize a estrutura de dados auxiliar que achar mais adequada para implementar a calculadora.
2. Escreva uma função que recebe como entrada o caminhamento em *pre-order* e *in-order* de uma árvore binária e exibe em `stdout` o caminhamento em *post-order* da árvore dada.
3. Escreva uma função que utiliza programação dinâmica para computar o valor de C_N , definido pela relação de recorrência

$$C_N = N + \frac{1}{N} \sum_{1 \leq k \leq N} (C_{k-1} + C_{N-k}), \quad \text{para } N \geq 1 \text{ com } C_0 = 1 \quad .$$

Use sua função para calcular o valor de C_{100} . Utilize programação dinâmica *top-down* ou *bottom-up*, como preferir.

4. Implemente uma função para ordenar um vetor contendo 100 *strings* (`char *`) de no máximo 17 caracteres. É proibido o uso da função de sistema `qsort` e das funções de manipulação de *strings* da biblioteca `string.h`. Você é livre para implementar o método de ordenação que quiser.