



Instituto Federal de Minas Gerais - Campus Ouro Branco

Curso: Sistemas de Informação

Disciplina: Arquitetura e Organização de Computadores

Professor: Saulo Henrique Cabral Silva

Trabalho Prático 3

Assembly

Valor: 16 pontos

Data da entrega: 31/08/2024

OBJETIVOS: Consiste em rever conceitos básicos de programação bem como a prática de desenvolvimento utilizando linguagem de baixo nível: *Assembly*.

DESCRIÇÃO: Seu José, um funcionário veterano da ferrovia, é conhecido por ser um dos últimos "trocadores de vagões" da região. Sua função única é reorganizar os vagões dos trens, garantindo que cada um seja descarregado na estação correta.

Em uma antiga estação ferroviária, onde a ponte girava 90 graus para permitir a passagem de barcos, Seu José descobriu que podia trocar a posição de dois vagões utilizando a ponte. Com um giro de 180 graus, os vagões ficavam invertidos, mas isso não era problema, já que os vagões podem se mover em ambas as direções.

Com a modernização da ferrovia, Seu José se viu diante de um novo desafio: automatizar seu trabalho. A empresa ferroviária deseja criar um programa que determine o número mínimo de trocas de vagões adjacentes necessárias para ordenar um trem.

SUA MISSÃO: Ajude Seu José a criar uma rotina que, dado um trem com vagões desordenados, determine o número mínimo de trocas de dois vagões adjacentes necessárias para ordená-lo.

ENTRADA:

1 - A entrada contém na primeira linha o número de casos de teste (N).

2 - Cada caso de teste consiste em $M + 1$ linhas de entrada.

- A primeira linha de um caso de teste contém um inteiro L, determinando a quantidade de vagões.
- As M linhas seguintes contém uma permutação dos números de 1 a L, indicando a ordem atual dos vagões. Os vagões devem ser ordenados de forma que o vagão 1 venha primeiro, depois o 2, etc., com o vagão L vindo por último.

EXEMPLO:

3

3

1

3

2

4

4

3

2

1

2

2

1

1 Troca

3 Trocas

1 Troca

O que deve ser entregue:

1. Código fonte do programa em **Mips/Assembly** (simulador *Mars* disponível no moodle).
2. Documentação do trabalho. Entre outras coisas, a documentação deve conter:
 - 2.1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
 - 2.2. Implementação: descrição sobre a implementação do programa. Deve ser detalhada a estrutura de dados utilizada (de preferência com diagramas ilustrativos), o funcionamento das principais funções e procedimentos utilizados, 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.
 - 2.3. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
 - 2.4. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet se for o caso
3. Formato: mandatoriamente em PDF (<http://www.pdf995.com/>).

Obs1: Apesar desse trabalho ser bem simples, a documentação pedida segue o formato da documentação que deverá ser entregue nos próximos trabalhos.

Obs2: Consulte as dicas do Prof. Nívio Ziviani de como deve ser feita uma boa implementação e documentação de um trabalho prático: <https://saulocabral.pagekite.me/roteirotp.pdf>

Como deve ser feita a entrega:

A entrega **DEVE** ser feita através da tarefa cadastrada na plataforma moodle, na forma de um único arquivo **zipado**, contendo o código, os arquivos e a documentação.

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar;
- Clareza, *indentação* e comentários no programa também vão valer pontos;
- O trabalho por ser feito em duplas (**grupo de DOIS alun@s**);
- Trabalhos copiados (e **FONTE**) terão nota ZERO;
- Trabalhos entregue em atraso serão aceitos, todavia a nota atribuída ao trabalho será zero
- Evite discussões inócuas com o professor em tentar postergar a data de entrega do referido trabalho.