1001350: Terceiro trabalho

Jander Moreira*

Entrega: 5 de junho de 2019

1 Introdução

Listas são uma importante ferramenta em computação, servindo de suporte a diversas estruturas de dados e proporcionando um recurso para o armazenamento de dados simples ou complexos.

Neste trabalho, uma versão básica de manipulação de listas ajudará na avaliação do aluno quanto ao uso de representação de dados, uso de subrotinas (procedimentos e funções) e passagem de parâmetros e conceitos de escopo de dados e declarações.

2 Especificação

Utilizando os recursos de estruturação de algoritmos, o desenvolvimento por refinamentos sucessivos e os conceitos de modularização, estruture uma solução para a manipulação de listas. Sua solução deve ser apresentada na forma de um programa escrito na linguagem C.

2.1 As listas

No contexto deste trabalho, uma lista é uma sequência de valores inteiros. Uma lista pode ser vazia (ou seja, ter comprimento zero), porém é finita (possui um número finito de itens).

São exemplos de listas:

- $\langle\rangle$ – sequência vazia, com comprimento zero;
- $\langle 1, 2, 3\rangle$ – sequência com três itens;
- $\langle 1, 1, 1\rangle$ – sequência com três itens, com repetição;
- $\langle -5, 8, 4, 17\rangle$ – sequência de comprimento quatro, sem ordenação.

A ordem relativa dos itens em uma sequência é relevante. Assim $\langle 1, 2, 3\rangle \neq \langle 3, 2, 1\rangle$, bem como $\langle 1, 2\rangle \neq \langle 1, 1, 2\rangle$.

Listas ordenadas são de especial interesse para a computação, por uma série de aplicações possíveis.

2.2 O problema geral

Devem ser estruturadas rotinas para a manipulação de listas ordenadas.

Em particular, deve ser possível:

- Verificar se uma lista qualquer está ordenada;
- Acrescentar um novo item em uma lista;
- Concatenar duas listas;
- Intercalar duas listas.

Uma lista $\langle s_1, s_2, \ldots, s_n\rangle$ é dita **ordenada** se $s_i \leq s_{i+1}$ para todos os valores válidos de i.

O **acréscimo** de um item a uma lista pode se dar sob dois pontos de vista diferentes: (a) acréscimo no final ou (b) acréscimo ordenado. Dado um novo item k e uma lista $\langle s_1, s_2, \ldots, s_n\rangle$, a inserção no final resulta na lista $\langle s_1, s_2, \ldots, s_n, k\rangle$, enquanto a inserção ordenada resulta em $\langle s_1, s_2, \ldots, s_{n+1}\rangle$, de forma que k é um dos valores s_i, mantendo a lista ordenada. No caso da inserção no final, o resultado pode não ser uma lista ordenada.

Dadas duas listas $\langle a_1, a_2, \ldots, a_n\rangle$ e $\langle b_1, b_2, \ldots, b_m\rangle$, a **concatenação** da listas resulta na lista $\langle a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m\rangle$. O resultado pode não ser uma lista ordenada.

Por fim, a intercalação de duas listas ordenadas $\langle a_1, a_2, \ldots, a_n\rangle$ e $\langle b_1, b_2, \ldots, b_m\rangle$ resulta em uma lista $\langle s_1, s_2, \ldots, s_{n+m}\rangle$, sendo os itens s_i uma permutação dos itens a_k e b_j quem mantém a ordenação total.

2.3 O programa

Deve ser desenvolvido um programa em C que permita a manipulação de listas de inteiros.

O programa deve realizar os seguintes passos:

- Ler uma sequência de inteiros e montar a primeira lista;
- Ler outra sequência de inteiros e montar a primeira lista;
- Se ambas as sequências forem ordenadas, gerar uma terceira lista pela intercalação das listas; caso contrário, a terceira lista deve ser a concatenação da primeira com a segunda;
- Apresentar a terceira lista.

Cada uma das duas sequências de entrada será fornecida primeiramente indicando o número de itens ($n \geq 0$), seguido dos valores s_1 a s_n.

*Moreira, J. – Universidade Federal de São Carlos – Departamento de Computação – *Rodovia Washington Luis, km 235 – 13565-905 - São Carlos/SP – Brasil* – jander@dc.ufscar.br

A apresentação da lista deve ser feita pela escrita de cada valor individual, separados por espaço simples, todos em uma única linha (i.e., sem `'\n'` entre os valores).

Deve ser escolhida uma estrutura de dados adequada para o armazenamento das listas. Caso seja necessário, pode-se assumir que uma lista nunca possuirá mais que 5000 itens.

Todas as manipulações de listas devem ser feitas por sub-rotinas, de forma que o uso das listas fique o mais separado possível de sua estruturação interna. Desta forma, por exemplo, a concatenação de duas listas deve ser feita por uma rotina que pegue duas listas como parâmtros e retorne a lista concatenada (parâmetro por referência ou como retorno de função). Da mesma forma, a verificação se uma lista está ordenada deve ser feita por uma função específica para esse fim. Mesmo a criação da lista com os dados lidos e a escrita devem ser feitas por sub-rotinas.

Aspectos de desempenho são importantes. Por exemplo, intercalar duas listas varrendo ambas simultaneamente e escolhendo de qual delas pegar o próximo item tem eficiência significativamente maior que concatenando duas listas e depois ordenando todos os dados.

Restrições

Será invalidada qualquer submissão que:

- Use, sob qualquer pretexto, os comandos `continue` ou `break` ou a função `exit` (ou similar);
- Use qualquer variável[1] global.

3 O que, quando e como entregar

O trabalho deve ser escrito como um único programa na linguagem C, sendo enviado apenas o código fonte. A submissão deve ser feita em http://aulas.dc.ufscar.br/~jander. Uma bateria de casos de teste será realizada e os resultados serão apresentados. O número de submissões é livre e cada nova versão substitui a antiga. Depois da data de entrega, novas submissões serão rejeitadas e a última versão enviada será a considerada para avaliação.

4 Critérios de avaliação

O código enviado será avaliado segundo os critérios e pesos indicados na Tabela 1. A **documentação** envolve tanto a documentação geral (entradas e saídas) quanto os comentários no código fonte; as sub-rotinas também devem ser devidamente comentadas. A **qualidade do código** se refere à estruturação da solução, uso de identificadores significativos, correta indentação etc. A qualidade das **sub-rotinsa** se refere ao uso adequado dos parâmetros, uso correto do escopo de definições e variáveis. Finalmente, a **execução dos casos de teste** são diretamente obtidas da submissão e correção automática realizadas.

Tabela 1: Critérios e pesos para avaliação.

Critério	**Peso**
Documentação	1,0
Qualidade geral do código	2,0
Qualidade das sub-rotinas	2,0
Execução dos casos de teste	5,0

5 Pontos importantes

São pontos importantes para a elaboração e entrega do trabalho:

- O trabalho deve ser desenvolvido individualmente e seguir as especificações;
- Qualquer ajuda externa deve seguir o *Código de conduta* disponibilizado para a disciplina;
- O código será avaliado pelo MOSS[2];
- Havendo qualquer dúvida, consulte o professor.

Atualizado em 8 de março de 2019.

[1] Declarações de tipos podem ser globais.

[2] http://theory.stanford.edu/~aiken/moss.