Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

SCC-602 Algoritmos e Estruturas de Dados 1 – (Engenharia de Computação)

Profª. Maria das Graças Volpe Nunes

Trabalho Prático 2

Data de divulgação: 14/10/10
Prazo para entrega: 05/11/10

Descrição do Problema

Criar um programa em linguagem C que implemente certas operações sobre listas dinâmicas simplesmente encadeadas, como segue:

- O programa deve permitir a manipulação de duas listas dinâmicas L1 e L2.
- As listas contêm valores inteiros que podem variar de 1 a 99 e aceitam valores repetidos.
- Ao ser iniciado, o programa exibe as duas listas (que são inicialmente vazias) e define L1 como a lista ativa. A lista ativa é aquela na qual deve ser realizado o próximo comando do usuário. Uma sugestão de apresentação é a seguinte:

➔ L1:
L2:

Isso indica que ambas as listas estão vazias (i.e., sem conteúdo) e que L1 é a lista ativa.

- A qualquer momento, o programa deve permitir que o usuário alterne a lista ativa (i.e., de L1 para L2 e vice-versa);
- O programa deve permitir a realização de operações sobre a lista ativa;
- As operações devem ser oferecidas ao usuário através de um menu, que será exibido após cada operação. Esse menu deve conter as seguintes opções:
    1. Alternar lista ativa
    2. Inserção
    3. Exclusão
    4. Preenchimento
    5. Intercalação
    6. Balanceamento
    7. Finalizar

Operações

Inserção

Esta operação solicita ao usuário um valor inteiro para inserção na lista ativa. A posição de inserção depende da ordem dos elementos na lista, pois a mesma deve ser mantida ordenada. Valores já existentes devem ser aceitos.

Exclusão de Intervalo

Esta operação solicita ao usuário dois valores inteiros N1 e N2 e remove da lista ativa todos os elementos cujas chaves estejam nesse intervalo (incluindo as chaves N1 e N2, se existirem).

Preenchimento

Esta operação solicita ao usuário dois valores inteiros N1 e N2 e percorre a lista ativa em busca de intervalos entre as chaves existentes e realiza a inserção de elementos intermediários a N1 e N2 (inclusive N1 e N2). Por exemplo, fornecidos os valores 5 e 10, o programa deverá inserir, na lista ativa os valores entre 5 e 10, mantendo a lista ordenada.

Lista ativa:

1 2 6 6 8 11

Na lista acima, a operação de preenchimento deve inserir novas chaves, deixando a lista com a seguinte configuração:

1 2 5 6 6 7 8 9 10 11

Intercalação (Merging)

Nesta operação, o programa deve realizar a intercalação de duas listas produzindo, colocando o resultado na lista ativa. A outra lista não é alterada. Exemplo:

➔ L1: 1 2 3 5 7 10
L2: 2 4 6 7 8 15

Resultado da intercalação:

➔ L1: 1 2 2 3 4 6 7 7 8 10 15
L2: 2 4 6 7 8 15

Balanceamento

Esta operação deve transferir elementos da lista maior para a menor até que fiquem com a mesma quantidade de elementos ou com a diferença de um. Por exemplo:

L1: 1 8 34

➔ **L2: 2 4 6 7 8 15 20 23**

Resultado do balanceamento:
L1: 1 2 6 8 34
➔ **L2: 4 7 8 15 20 23**

Fim
Esta opção simplesmente encerra o programa.

- Após cada operação o programa deve exibir o conteúdo atual de L1 e L2 (indicando a lista ativa) e esperar um novo comando.

Ferramentas

A implementação do trabalho será em linguagem C, utilizando o compilador GCC.

Diversos ambientes de programação utilizam o GCC como compilador padrão, com é o caso do Dev-C++ (http://dev-c.softonic.com.br/) e o Code Block (http://www.codeblocks.org/), inclusive esse compilador é o padrão da maioria das distribuição Linux.

Caso deseje instalar esse compilador para ser usado em linha de comando nos sistemas Windows, baixar o MinGW (http://www.mingw.org/, baixar em http://sourceforge.net/projects/mingw/).

Entrega do Trabalho

Deve ser entregue, até a data determinada, um arquivo zipado chamado GrupoXX.zip, onde XX corresponde ao número do seu grupo, contendo os arquivos abaixo:

- **principal.c** (Código onde se faz a interação com o usuário);
- **listas.h** (*Header* do TAD da implementação das operações sobre listas dinâmicas);
- **listas.c** (TAD da implementação das operações sobre listas dinâmicas);

A entrega do arquivo deve ser pelo email arquivos.maziero@gmail.com. Serão considerados os email enviados até às 23:59 do dia 29/10/2010. Trabalhos fora do prazo serão penalizados (como descrito abaixo).

Critérios de Avaliação e Penalidade

A nota do trabalho 1 (NT1) será constituída da seguinte maneira:

NT1 = 0.5*NC + 0.2*NE + 0.1*NI + 0.2*NCF, onde:

NC é a nota do critério Correção;

NE é a nota do critério Eficiência;

NI é a nota do critério Interface;

NCF é a nota do critério Código Fonte.

• **Correção (NC)**: o programa faz o que foi solicitado? Faz tudo o que foi solicitado?

Utiliza encapsulamento de informação? Serão considerados nos critérios de avaliação:

- O uso correto dos TAD e operações de manutenção relacionadas;
- Cumprimento dos requisitos funcionais, ou seja, realiza as transações da maneira correta;

• **Eficiência (NE)**: as operações são executadas da maneira mais eficiente para cada estrutura de dados? Evita código duplicado/redundante/não atingível?

• **Interface (NI)**: é simples de usar, prático, tolera os erros mais óbvios? O trabalho foi entregue dentro das especificações (zipado, com os nomes de arquivo solicitados)?

- Interface do programa;
- Interface de entrega do trabalho;

• **Código fonte (NCF)**: é claro e organizado? Nomes de variáveis são sugestivos? Está bem documentado?

- Clareza;
- Nomes de variáveis;
- Documentação/comentários no código.

A nota do trabalho sofrerá penalidade se entregue depois do prazo estipulado, conforme os critérios abaixo:

NT1 = k***NT1**, onde k é um fator multiplicador dado por:

k = 1.0, se o trabalho for entregue dentro do prazo determinado (não há penalidade);

k = 0.7, se o trabalho for entregue até 24hs após o prazo determinado;

k = 0.5, se o trabalho for entregue entre 24hs e 48hs após o prazo estipulado;

k = 0.0, se o trabalho for entregue mais de 48hs após o prazo estipulado (será considerado desistente).

Para cada trabalho recebido por e-mail será enviada uma resposta de confirmação!