SCC 0503 – Algoritmos e Estruturas de Dados II (Sistemas de Informação)
SCC 0603 – Algoritmos e Estruturas de Dados II (Engenharia da Computação)

Professora: Elaine Parros Machado de Sousa – parros@icmc.usp.br
Estagiários PAE: Diego Silva – diegofsilva@icmc.usp.br
Ivone Penque Matsuno – ivone.matsuno@usp.br

Trabalho 2 – Arquivos

1. Objetivo do Trabalho:

Aplicar os conceitos estudados na disciplina de Algoritmos e Estrutura de Dados II para implementação de soluções para problemas práticos.
Neste trabalho, o aluno deverá desenvolver um programa em linguagem C ou C++ para resolução do problema especificado na **seção 4**.
O programa deve ser compilado no **gcc/g++ 4.2.7.** ou no **codeblocks 13.11**.

2. Critérios de Avaliação

O programa deve ser desenvolvido em grupo de no máximo 3 alunos e deve seguir rigorosamente os formatos de entrada e saída definidos. Quaisquer programas similares (parcialmente ou totalmente) terão **nota zero**, independente de qual for o original e qual for a cópia. Programas desenvolvidos em outras linguagens (diferentes de C/C++) não serão aceitos.

- A nota atribuída ao trabalho será de zero a dez inclusive.
- A nota máxima é atribuída se o trabalho avaliado atender a todos os requisitos e o trabalho estiver bem organizado e apresentado.
- Para cada requisito não atendido no trabalho, será descontada uma pontuação da nota até o limite mínimo da nota.
- Será atribuída nota ZERO quando:
    - o trabalho não for submetido até a data máxima possível;
    - ou/e o trabalho apresentar muitos erros;
    - ou/e não compilar;
    - ou/e for detectado plágio do trabalho (parcial ou total);
    - ou/e, não for entregue relatório conforme especificação.

Os itens de avaliação dos programas são:

1. (30%) O programa funciona corretamente nas tarefas descritas para todos os casos de teste e processamento correto das entradas e saídas;
2. (40%) Qualidade da solução desenvolvida, modularização do código, bom uso das técnicas de programação e eficiência da solução em termos de espaço e tempo;
3. (10%) Boa indentação e uso de comentários no código, além de boa estruturação e modularização;
4. (20%) Documentação externa: **relatório** curto (em pdf) explicando cada decisão tomada para a implementação da solução, sendo necessária a inclusão de detalhes sobre (pelo menos) os tipos de registro e campo, estruturas de dados utilizadas, estratégias de implementação das operações de busca, inserção e remoção, estratégias de alocação de *slots* (se for o caso), estratégias de indexação (se for o caso) e estimativas de complexidade de tempo e espaço. Inclua figuras para ilustrar a organização do arquivo e as estruturas de apoio (nos moldes das figuras apresentadas em aula).

3. Data e Forma de Entrega

Data de entrega: 25/05/2015 (2a-feira)

A data de entrega será considerada a da última submissão dos arquivo no Tidia-ae.

A cada dia de atraso será descontado 1 ponto da nota final do trabalho. Serão considerados os dias consecutivos independentemente se é dia útil ou não.

Após encerrado o prazo máximo de 10 dias de atraso, não serão aceitos mais trabalhos.

Forma de entrega:
A entrega será realizada no ambiente Tidia-ae na Atividade **Trabalho 2**.

O que entregar?
Você deve entregar os arquivos contendo **apenas o seu programa fonte e o relatório**. A pasta completa do projeto deve ser compactada em um único arquivo (com extensão “zip”).

O nome do arquivo compactado deve ser formado pela sigla “T2-” concatenada à 1ª letra do nome concatenada ao último sobrenome de um dos integrantes do grupo.
Exemplo: T2-imatsuno, T2-dsilva, T2-esousa, etc.

4. Descrição do Problema

Você foi contratado para desenvolver um sistema para um site que oferece serviços de *streaming* de música. Esse sistema ajudará a gerenciar as preferências dos usuários, para que isso possa ser utilizado em sistemas de recomendação. Esse site possui um arquivo em que armazena os seguintes dados (nessa ordem):

1. ID numérico do usuário
2. Nome completo do usuário
3. Idade
4. Sexo: masculino (M) ou feminino (F)
5. Gêneros musicais preferidos
6. Tipo de usuário (TU)

No campo 5 são considerados todos os gêneros que o usuário afirma gostar, como por exemplo, heavy metal, classic rock, hip-hop, sertanejo, etc. No caso em que o usuário não tenha escolhido nenhum gênero, o campo deve conter o valor “unknown”. Os gêneros estão separados por arroba (@).

O campo tipo de usuário (TU) representa uma política do site para oferecer serviços e propaganda para os usuários. Os possíveis TU são: gratuito, comum e premium.

O site quer um sistema que armazene seus dados do melhor modo para assim poder fazer consultas **eficientes** e recomendações a seus clientes. As **funcionalidades** requeridas são:

1. Inserção de novos usuários
2. Remoção de usuários
3. Pesquisa por ID
4. Dado um conjunto não vazio (G) de gêneros, o dono do site quer:
    a. Os 3 gêneros musicais mais populares entre as pessoas que compartilham o gosto pelos gêneros em G (excluindo os gêneros contidos em G).

    b. Os IDs das 10 pessoas mais jovens (caso haja menos, retornar todos) de um tipo de usuário (TU), que compartilham o gosto pelos gêneros em G.
5. Dada uma faixa de idade:
    a. Retornar (até) os 10 gêneros mais populares para essa faixa
    b. Dado um gênero, listar o ID das pessoas que gostam daquele gênero e têm a idade dentro da faixa de idade indicada

O programa deve permitir a interação pelo console/terminal (modo texto). Os requisitos funcionais do programa, necessários para realização das demais operações, são:

- Verificar se existem arquivos de índices e de dados. Se não existirem índices, mas existirem os arquivos de dados, então criar os índices.
- Verificar se os índices estão atualizados.
- Assuma que os arquivos de dados encontram-se no diretório raiz do programa.
- O programa deve fazer tratamento/validação de dados informados, como ID, sexo, idade e tipo de usuário.
- A entrada de dados será ASCII e não serão considerados caracteres especiais (~, ç, ^, `, ’, etc.).

Importante: Projete o programa de modo a ser eficiente, com um potencial de funcionalidade para uma carga de no máximo 100.000 registros.

5. Entrada

A interação com o usuário por meio do console/terminal, deve seguir um formato padrão de entrada. Essas informações serão fornecidas no formato descrito a seguir.

O programa deve iniciar com a leitura de um código numérico inteiro C, indicando a operação a ser realizada (listada num menu). Considere os seguintes códigos de operação:

1. Inserir usuário (funcionalidade 1)
2. Remover usuário (funcionalidade 2)
3. Pesquisar por ID (funcionalidade 3)
4. Buscar gostos musicais semelhantes, dado um conjunto de gêneros (funcionalidade 4.a)
5. Buscar usuários mais jovens, dado um tipo de usuário e um conjunto de gêneros (funcionalidade 4.b)
6. Buscar gêneros mais populares, dada uma faixa de idade (funcionalidade 5.a)

7. Buscar usuários de uma faixa de idade que gostam de um determinado gênero (funcionalidade 5.b)
8. Fechar o programa

Caso a operação escolhida seja a de código **1**, seu programa deve esperar, nessa ordem (e separados por \n):

- Número inteiro com ID do usuário
- Nome completo do usuário
- Número inteiro com a idade do usuário
- Caracter que determina o sexo do usuário
- Uma *string* com os gêneros musicas separados pelo caracter '@'
- Número inteiro indicando o tipo de usuário (1 para gratuito, 2 para comum e 3 para premium)

Caso a operação escolhida seja a de código **2**, seu programa deve ler apenas um número inteiro discriminando o ID do usuário a ser removido. O mesmo vale para a operação de código **3**, em que o ID lido será buscado no arquivo.

Caso a operação possua código **4** ou **5**, seu programa deve ler uma *string* de gêneros musicais (separados por '@') referentes à consulta. Adicionalmente, para a operação **5**, seu programa deverá ler um número inteiro para identificar o tipo de usuário que a busca irá abranger. Nesse caso, o seu programa deve considerar quatro opções: 1 - gratuito, 2 - comum, 3 - premium e 4 - qualquer tipo de conta.

Finalmente, para as operações **6** e **7**, seu programa deve ler dois números inteiros determinando o início e o fim da faixa de idade considerada. Adicionalmente, para a operação de código **7**, seu programa deve ler uma *string* que determina o gênero relativo à busca.

Após cada operação (exceto a de código 8), o programa deve retornar à leitura do código de operação **C**.

6. Saída

Ao executar as operações **1** e **2**, seu programa deve indicar ao usuário se a operação falhou ou foi finalizada com sucesso. Para o segundo caso, imprima a mensagem "Operacao realizada com sucesso". Caso a operação não possa ser efetuada, as falhas

para as operações de código **1** e **2** devem ser “Falha: ID ja cadastrado” e “Falha: ID nao encontrado”, respectivamente. Essa mensagens devem ser impressas na tela sem as aspas e terminando com o caracter ‘\n’.

Para a operação **3**, sua saída deve conter um campo por linha e nessa ordem:

- Número inteiro com ID do usuário
- Nome completo do usuário
- Número inteiro com a idade do usuário
- Caracter que determina o sexo do usuário
- Uma *string* com os gêneros musicas separados pelo caracter ‘@’
- Número inteiro indicando o tipo de usuário (1-gratuito, 2-comum e 3-premium)

A saída das operações **4** e **6** devem conter uma *string* por linha, determinando os gêneros musicais encontrados na busca. Analogamente, as operações **5** e **7** devem imprimir na tela um número inteiro por linha, indicando o ID dos usuários encontrados.

ATENÇÃO

- Não deverá ser utilizada qualquer variável global.
- Não poderão ser utilizadas bibliotecas com funções prontas (a não ser aquelas para entrada, saída e alocação dinâmica de memória).
- Dúvidas conceituais deverão ser colocadas nos horários de atendimento.