Bacharelado em Ciência da Computação **DINF / UFPR**
CI067 - Oficina de Computação **1º semestre 2015**
Prof. Armando Luiz N. Delgado http://www.inf.ufpr.br/nicolui
TRABALHO 01 - TURMA K **26/03/2015**

Releases - Histórico:
r0 26/03/2015 Primeira versão

1 Especificação do Trabalho

Frederic Chopin foi um compositor polonês que viveu de 1810-1839. Um de seus mais famosos trabalhos foram seu conjunto de prelúdios. Estas 24 peças musicais cobrem as 24 teclas musicais (existem 12 notas musicais distintas, e cada uma pode ter a tonalidade maior ou menor). As 12 notas distintas da escala musical são:

1	2	3	4	5	6	7	8	9	10	11	12
$Lá$	$Lá\sharp = Si\flat$	Si	$Dó$	$Dó\sharp = Ré\flat$	$Ré$	$Ré\sharp = Mi\flat$	Mi	$Fá$	$Fá\sharp = Sol\flat$	Sol	$Sol\sharp = Lá\flat$

Cinco das notas tem dois nomes alternativos, como indicado acima com os sinais de igualdade (e.g. $Dó\sharp = Ré\flat$ significa que aquela nota tem dois nomes, $Dó\sharp$ e $Ré\flat$). Desta forma, existem 17 nomes possíveis para as notas da escala musical, mas somente 12 delas são musicalmente distintas. Quando se usa uma delas, nós podemos então distinguir entre as tonalidades maior e menor. Isto nos dá 24 teclas possíveis, das quais 24 são musicalmente distintas.

Ao nomear seus prelúdios, Chopin usou todas as teclas exceto as 10 a seguir:

$$\begin{array}{lllll} Lá\flat\ menor & Lá\sharp\ maior & Lá\sharp\ menor & Dó\sharp\ maior & Ré\flat\ menor \\ Ré\sharp\ maior & Ré\sharp\ menor & Sol\flat\ maior & Sol\flat\ menor & Sol\sharp\ maior \end{array}$$

Escreva um programa chamado preludios que, dado o nome da tecla, devolva um nome alternativo (se existir) ou indique que aquela tecla é única. Além disso, a saída deve também dar a resposta usando a notação musical ABC para as notas ($C = Dó, D = Ré, E = Mi, F = Fá, G = Sol, A = Lá, B = Si$)

1.1 Entrada e Saída de dados

A entrada consiste de linhas, cada uma contendo um texto no formato *nota tonalidade*, onde $nota$ é um dos 17 nomes na escala de notas acima, e $tonalidade$ é $maior$ ou $menor$. Todos os nomes de notas devem ser maiúsculas, e os dois acidentes ($\sharp$ e $\flat$) serão escritos como $\#$ e b, respectivamente. O final da entrada é indicada pelo texto *FIM*.

Para cada entrada, mostre o número da linha, seguida pelo nome alternativo da tecla, se existir, ou imprima a palavra *ÚNICO* se o nome de tecla é único. Siga o formato do exemplo abaixo.

Exemplo de Entrada	Exemplo da Saída
LA-b menor	Caso 1: SOL-# menor –> G# menor
RE-# maior	Caso 2: MI-b maior –> Eb maior
SOL menor	Caso 3: UNICO
FIM	

1.2 Execução do Programa

O pacote de software a ser construído consiste do programa preludios .

O programa deve ler sua entrada via teclado e a saída deve ser para a tela do computador. Redirecionamento de entrada pode ser usada para utilizar os arquivos de exemplo de entrada aqui[1]

O programa não deve emitir quaisquer mensagens de referência para a entrada ou a saída de dados.

Se algum dos dados de entrada não satisfizer as restrições indicadas na seção 1.1, o programa deve terminar com a mensagem PARÂMETROS FORA DE ESPECIFICAÇÃO.

[1] Arquivos com extensão **.in** são exemplos de entrada e arquivos com extensão **.out** mostram o resultado que o programa deve emitir na tela.

2 Produto a ser Entregue

O trabalho deve ser desenvolvido INDIVIDUALMENTE.

O aluno deve entregar um pacote de software completo contendo os fontes em linguagem C dos programas solicitados e documentação. O pacote deve ser arquivado e compactado com ***tar(1)*** e ***bzip2(1)*** em um arquivo chamado login1.tar.bz2, onde login1 é *login* do aluno nos sistemas computacionais do DINF/C3SL.

O pacote deve ter a seguinte estrutura de diretórios e arquivos:

- ./login1/: diretório principal. Ao ser executado, o programa devem assumir que este é o diretório corrente;
- ./login1/LEIAME;
- ./login1/src/: diretório contendo todos os arquivos fonte em linguagem C (*.c);

Note que a extração dos arquivos em login1.tar.bz2 deve criar o diretório login1 contendo todos os arquivos e diretórios acima.

2.1 Documentação

O pacote deve conter um arquivo de documentação em texto plano (ASCII). Este arquivo, chamado LEIAME, deve conter as seguintes informações:

- autoria do software, isto é, nome e RA dos membros do grupo;
- lista dos arquivos e diretórios contidos no pacote e sua descrição (breve);
- um capítulo especial descrevendo os algoritmos e as estruturas de dados utilizadas, as alternativas de implementação consideradas e/ou experimentadas e os motivos que o levaram a optar pela versão entregue, as dificuldades encontradas e as maneiras pelas quais foram contornadas.
- *bugs* conhecidos;
- outras informações que forem julgadas importantes.

2.2 Arquivos Fonte

O programa deve ser implementado exclusivamente em linguagem C , e deve ser composto de no mínimo: a função **main()**, e no mínimmo mais 2 funções que devem ser criadas e usadas pelo programa como parte do algoritmo completo que resolve o problema.

O programa deve estar em um arquivo de nome **preludios .c**.

3 Entrega

A entrega do trabalho será feita em 3 (três) fases:

Fase 1 - 26/03/2015, 19:00h : O aluno deve entregar o programa com seu esqueleto definido em pelo menos 90%. Isto inclui definição de tipos de dados, constantes, funções e o código da função main() indicando a chamadas das funções e a estrutura geral do programa. As funções deverão OBRIGATÓRIAMENTE ter um comentário indicando o objetivo da função, descrição dos parâmetros e dos valores de retorno, se houver, seu cabeçalho e protótipo.

Fase 2 - 29 de março de 2015, 22:30H: O aluno deve enviar o programa na forma e estado em que estiver. Não precisa estar funcionando ou compilando corretamente. Esta fase serve para o professor verificar até que ponto o aluno chegou no desenvolvimento. A única restrição é que o código entregue nesta fase pode diferir do que foi entregue na Fase 1 em até 20%. Novas funções podem ser criadas, mas estas novas funções não podem substituir as definidas na Fase 1.

Fase 3 - 31 de março de 2015, 19:30H: O aluno deve ter seu programa compilando e executando e usar o tempo de aula para corrigir falhas e responder aos questionamentos que o professor fará durante a aula. ALUNOS QUE NÃO COMPARECEREM NESTA AULA TERÃO NOTA 0 (ZERO). Ao final da aula, o programa deverá ser entregue no estado en que estiver e será considerado o estado final do programa para correção pelo professor.

ATENÇÃO:	O não cumprimento de qualquer de uma das fases acima implicará em nota 0 (zero) para o aluno.

O trabalho deve ser enviado como anexo por e-mail ao professor responsável pela disciplina:

- A mensagem com o anexo deve ser enviada ao Prof. de sua turma: Turma A (Aldri Santos <ci067ufpr@gmail.com>) ou Turma K (Armando Delgado <nicolui@inf.ufpr.br>), com o Assunto (*Subject*): **CI067 - Trabalho 01 2015**.
- No corpo da mensagem DEVE CONSTAR OBRIGATORIAMENTE o Nome e Número de Registro Acadêmico (RA) do aluno autor do trabalho;
- O aluno deverá considerar o trabalho como entregue SOMENTE APÓS RECEBER DO PROFESSOR UMA MENSAGEM DE CONFIRMAÇÃO DE RECEBIMENTO dentro de 24 horas desde o envio da mensagem;

4 Critério de Avaliação

APENAS OS TRABALHOS QUE COMPILAREM SEM ERROS SERÃO CORRIGIDOS. Se o trabalho não compilar ou acusar falha de segmentação (*Segmentation fault*) prematura durante os testes realizados pelo professor (sem que qualquer operação se efetue a contento), trará para o aluno NOTA 0 (ZERO). Também receberão NOTA 0 (ZERO) trabalhos plagiados de qualquer fonte, e/ou com códigos idênticos ou similares. Além disso, apenas trabalhos entregues no prazo marcado receberão nota.

Legibilidade, elegância, eficiência, modularidade, coesão e acoplamento entre módulos, e uso adequado de estruturas de dados serão levados em conta na avaliação.

Os algoritmos de manipulação das estruturas de dados (inserção, ordenação, etc.) devem ser tão eficientes quanto possível, usando todos os conceitos vistos nas disciplinas de Algoritmos do Curso. Estruturas dinâmicas abordadas em Alg II, devidamente utilizadas, colaboram fortemente para uma avaliação positiva.

Caso seu programa possua *bugs* conhecidos, descreva-os no arquivo LEIAME. A documentação de um *bug* pode evitar que o aluno receba nota Zero.

Os itens de avaliação do trabalho e respectivas pontuações são:

Qualidade da documentação: 15 pontos

Qualidade do código: 25 pontos

Implementação: 60 pontos

Defesa: A defesa do trabalho será oral, e definirá a nota individual de cada aluno, de acordo com seu conhecimento a respeito do trabalho.

O item Qualidade da documentação se refere ao arquivo LEIAME. O item Qualidade do código inclui os comentários no código fonte e grau de portabilidade do código. O item Eficiência da implementação inclui análise de estruturas de dados, de algoritmos utilizados e sua eficiência.

5 Casos Omissos

Quaisquer dúvidas a respeito da especificação, entrega ou avaliação do trabalho deverão ser encaminhadas aos professores da disciplina, pessoalmente ou através de mensagens eletrônicas.