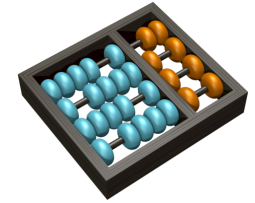


Instituto de Computação - UNICAMP

MC102 - Algoritmos e Programação de Computadores



Laboratório 10: Autômato musical

Segundo Semestre de 2017 - Turmas Coordenadas

Peso da Atividade: 2

Prazo de Entrega: 19 de novembro de 2017 às 23:59:59

Conteúdo

[Contexto](#)

[Tarefa](#)

[Exemplos](#)

[Observações da Tarefa](#)

[Observações Gerais](#)

[Critérios Importantes](#)

[Para Curiosos Interessados](#)

Desativar reprodução

Contexto

***"Jeune homme, vous commencez par où je voudrais
finir."***

– Jacques de Vaucanson

Les 3 automates Jaquet-Droz



Em toda Suíça, não há melhor relojoaria do que aquela em que você trabalha. Vocês fabricam os mecanismos mais precisos e avançados. Agora, vocês estão preparando mais uma maravilha de inovação tecnológica para aumentar a venda de seus famosos relógios-cuco. Ano passado, em 1793, Johann Julius Hummel publicou em Berlim [Anleitung so viel Walzer oder Schleifer mit zwei Würfeln zu componiren so viel man will ohne musikalisch zu seyn noch etwas von der Composition zu verstehe](#), cuja autoria é atribuída a Wolfgang Amadeus Mozart. Como o título diz, a obra permite que pessoas sem conhecimento algum na área de composição musical componham tantas valsas ou ländler quantos queiram. É claro que vocês não podiam deixar passar a oportunidade de incorporar isto na sua próxima criação, o autômato **l'Orchestre à Ressort**, que pretendem dar de presente ao Imperador Franz II, por ocasião do seu trigésimo aniversário.



Brasão do sacro imperador romano-germânico Franz II, ladeado de grifos rampantes, em formato SVG.

A maravilha mecânica que vocês estão construindo produzirá sempre músicas novas, com sons diversificados, por meio da permutação do uso de cilindros com linguetas. **L'Orchestre à Ressort** é um complexo de 128 instrumentos, que podem produzir, cada um, sons variados, em até 128 notas distintas e 127 nuances de intensidade distintas. Os instrumentos são acionados por bonecos antropomórficos controlados por alavancas e engrenagens. É o movimento dos dedos dos bonecos que aciona o início e o fim de cada som, dando a impressão de que sejam músicos de orquestra reais. Alguns colegas seus estão trabalhando na parte de aleatorização. Outros estão trabalhando nos instrumentos. Seu papel será modelar o funcionamento da parte que traduz as informações, dos sons a serem produzidos, em *eventos* que controlam os dedos dos bonecos. Há movimentos que iniciam os sons nos instrumentos e movimentos que cessam esses sons.

O mecanismo que gera todo o projeto para a música termina seu funcionamento antes que qualquer som seja emitido. Tal mecanismo monta um conjunto de sons que deverão ser tocados ao longo da música em cada momento. Cada som tem seu *instrumento*, sua *nota* (de mais grave a mais aguda), sua *duração* (de mais curta a mais longa) e sua *intensidade* (de mais fraca a mais forte em volume sonoro). As informações são transmitidas aos bonecos na forma de *eventos* de dois tipos: início ou fim de som. O instrumento deve soar a nota especificada pelo evento que o iniciou até que outro evento o silencie.

Tarefa

Implementar um programa que:

- receba, como entrada, um conjunto de sons a serem produzidos;
- produza, como saída, os eventos **ordenados** que permitirão a realização mecânica da música.

A seguir, a entrada e a saída do programa estão detalhadas.

Entrada

Cada linha contém as informações para um som a ser produzido ao longo da música, na forma de cinco inteiros separados por espaços, nesta ordem:

- O número (de 1 a 128) do instrumento que deve produzir o som,
- O número (de 0 a 127) da nota que o som deve ter,
- A intensidade (de 1 a 127) com que o som deve soar,
- O instante de tempo em que o som deve ser iniciado e
- A duração de tempo em que som deve ser mantido.

Apenas a última linha contém cinco números zero e não representa um som, mas indica o fim da entrada.

Saída

Para gerar a saída, você deverá chamar uma única vez a função `saida(int modo, int numEventos, TpEvento eventos[])`, em que `modo` deve conter o valor zero, `numEventos` é o número de eventos total da instância. Estes eventos estão em um vetor `eventos` cujos elementos são do tipo `TpEvento`, definido a seguir.

```
typedef struct {  
    int instante;  
    int instrumento;  
    int nota;  
    int intensidade;  
} TpEvento;
```

A música se inicia no instante zero. Para cada som representado na entrada, deve haver dois eventos:

- um em que `instante` guarda o instante *inicial* do som e `intensidade` a intensidade do som; e
- outro em que `instante` guarda o instante *final* do som e `intensidade` vale zero, representando o fim do som.

Eventos que ocorram antes na música também devem estar em posições anteriores no vetor. Entre eventos simultâneos na música, eventos com número de instrumento menor, devem estar em posições anteriores no vetor. Entre eventos simultâneos no mesmo instrumento, eventos com número de nota menor devem estar em posições anteriores no vetor. No máximo dois eventos podem ter valores iguais para os três parâmetros: *instante*, *instrumento* e *nota*. Um deles termina um som anterior e outro inicia um som posterior. Nesses casos, o evento que termina o som anterior deve estar no vetor em posição anterior ao evento que inicia o som posterior.

A função *saida*, bem como o tipo *TpEvento* estão definidos na biblioteca *lab10.h*, que deve ser incluída como cabeçalho no seu programa.

Exemplos

Notas:

Textos em azul designam dados de entrada, isto é, que devem ser lidos pelo seu programa.
Textos em preto designam dados de saída, ou seja, que devem ser impressos pelo seu programa.

Exemplo de execução 1:

```
1 64 95 30 30
1 60 127 60 60
1 67 127 0 90
0 0 0 0 0
```

Instante	Instrumento	Nota	Intensidade
0	1	67	127
30	1	64	95
60	1	60	127
60	1	64	0
90	1	67	0
120	1	60	0

Sendo esta a entrada, o seu programa deve chamar `saida(0, 6, eventos)`, para que o conteúdo dos parâmetros dos elementos do vetor `eventos` sejam impressos na saída padrão formatado como acima.

Exemplo de execução 2:

```
3 59 127 0 24
3 56 127 36 24
2 59 64 12 36
2 52 64 48 12
0 0 0 0 0
```

Instante	Instrumento	Nota	Intensidade
0	3	59	127
12	2	59	64
24	3	59	0
36	3	56	127
48	2	52	64
48	2	59	0
60	2	52	0
60	3	56	0

Sendo esta a entrada, o seu programa deve chamar `saida(0, 8, eventos)`, para que o conteúdo dos parâmetros dos elementos do vetor `eventos` sejam impressos na saída padrão formatado como acima.

Exemplo de execução 3:

```
80 69 120 0 95
80 69 100 95 127
0 0 0 0 0
```

Instante	Instrumento	Nota	Intensidade
0	80	69	120
95	80	69	0
95	80	69	100

222	80	69	0
-----	----	----	---

Sendo esta a entrada, o seu programa deve chamar `saida(0, 4, eventos)`, para que o conteúdo dos parâmetros dos elementos do vetor `eventos` sejam impressos na saída padrão formatado como acima.

Observações da Tarefa

- Não há limite para o número de notas. Você deve alocar a memória dinamicamente.
- Toda memória alocada dinamicamente deve ser liberada antes do seu programa retornar.

Observações Gerais

- O número máximo de submissões é 20.
- O arquivo `lab10.c` deve conter todo o seu programa.
- Para a realização dos testes automáticos, a compilação se dará da seguinte forma: `gcc lab10.c -o lab10 -Wall -Werror -ansi -pedantic`.
- Não se esqueça de incluir no início do programa uma breve descrição dos objetivos, da entrada, da saída, seu nome, RA e turma.
- Após cada submissão, você deve aguardar um minuto até poder submeter seu trabalho novamente.
- Ao completar esta tarefa você terá aprendido a manipular registros.

CrITÉRIOS Importantes

O **não** cumprimento dos critérios abaixo acarretará em **nota zero na atividade**, independentemente dos resultados dos testes do SuSy.

- Sua solução deve atender todos os requisitos definidos no enunciado.
- Não serão aceitas soluções contendo estruturas não vistas em sala (para este laboratório, poderão ser utilizadas apenas variáveis simples, operações de entrada e saída, operações aritméticas, desvios condicionais, estruturas de repetição, vetores, matrizes, strings, funções, apontadores e registros).
- Não é permitido o uso de `continue` e `break` (exceto em estruturas do tipo `switch-case`).
- Não é permitido o uso de variáveis globais.
- Cada função deve conter apenas um único `return`.
- Os únicos cabeçalhos aceitos para inclusão são `lab10.h`, `stdlib.h` e `stdio.h`.
- As únicas funções permitidas da biblioteca `stdlib.h` são `calloc()`, `free()`, `malloc()` e `realloc()`.

Para Curiosos Interessados

Caso, na chamada à função `saida`, o parâmetro `modo` tenha o valor um em vez de zero, `saida` coloca na saída padrão o conteúdo binário de um arquivo MIDI contendo a música gerada. A pasta de arquivos auxiliares contém arquivos .MIDI referentes a casos de teste abertos. Os casos de teste 9 e 10 foram gerados a partir da publicação de Hummel. O código `entrada.c`, disponibilizado na pasta de arquivos auxiliares, gera instâncias aleatórias baseadas na obra atribuída a Mozart.

A figura abaixo representa a correspondência das notas na pauta musical, suas frequências sonoras fundamentais (em Hz) , as teclas do teclado (órgão, piano, etc), nome da nota em inglês (C = dô, D = ré, E = mi, F = fá, G = sol, A = lá, B = si) e seu número MIDI.

	Frequency	Keyboard	Note name	MIDI number
	4186.0		C8	108
	3951.1		B7	107
3729.3	3520.0		A7	106
3322.4	3136.0		G7	104
2960.0	2793.8		F7	102
	2637.0		E7	100
2489.0	2349.3		D7	99
2217.5	2093.0		C7	97
	1975.5		B6	95
1864.7	1760.0		A6	94
1661.2	1568.0		G6	92
1480.0	1396.9		F6	90
	1318.5		E6	88
1244.5	1174.7		D6	87
1108.7	1046.5		C6	85
	987.77		B5	83
932.33	880.00		A5	82
830.61	783.99		G5	80
739.99	698.46		F5	78
	659.26		E5	76
622.25	587.33		D5	75
554.37	523.25		C5	73
	493.88		B4	71
466.16	440.0		A4	70
415.30	392.00		G4	68
369.99	349.23		F4	66
	329.63		E4	64
311.13	293.67		D4	63
277.18	261.6		C4	61
	246.94		B3	59
233.08	220.00		A3	58
207.65	196.00		G3	56
185.00	174.61		F3	54
	164.81		E3	52
155.56	146.83		D3	51
138.59	130.81		C3	49
	123.47		B2	47
116.54	110.00		A2	46
103.83	97.999		G2	44
92.499	87.307		F2	42
	82.407		E2	40
77.782	73.416		D2	39
69.296	65.406		C2	37
	61.735		B1	35
58.270	55.000		A1	34
51.913	48.999		G1	32
46.249	43.654		F1	30
	41.203		E1	28
38.891	36.708		D1	27
34.648	32.703		C1	25
	30.868		B0	23
29.135	27.500	J. Wolfe, UNSW	A0	22

Produzido por [Joe Wolfe](#), Universidade de Nova Gales do Sul, Sydney, Austrália.

A tabela abaixo indica os instrumentos sugeridos com sua numeração segundo o padrão *General Midi*, de 1991, comumente empregado pela maior parte de reprodutores de arquivos com extensão **.midi**:

Número	Instrumento	Número	Instrumento
1.	Piano de cauda	65.	Saxofone soprano
2.	Piano brilhante	66.	Saxofone contralto
3.	Piano eletro-acústico	67.	Saxofone tenor
4.	Piano preparado	68.	Saxofone barítono
5.	Piano tipo Fender Rhodes	69.	Oboé
6.	Piano tipo DX7	70.	Corno inglês
7.	Cravo	71.	Fagote
8.	Clavicórdio	72.	Clarineta
9.	Celesta	73.	Flautim
10.	<i>Glockenspiel</i>	74.	Flauta transversal
11.	Caixinha de música	75.	Flauta doce
12.	Vibrafone	76.	Zamponha
13.	Marimba	77.	Gargalo de garrafa
14.	Xilofone	78.	<i>Shakuhachi</i>
15.	Carrilhão	79.	Assobio
16.	Dulcimer	80.	Ocarina
17.	Órgão Hammond	81.	Onda quadrada
18.	Órgão percussivo	82.	Onda dente de serra
19.	Órgão de rock	83.	Órgão a vapor
20.	Órgão de igreja	84.	Sopro sintético
21.	Harmônio	85.	Charanga sintética
22.	Acordeão	86.	<i>Solo vox</i>
23.	Sanfona	87.	Dente de serra com quintas paralelas
24.	<i>Bandoneón</i>	88.	Baixo e solo