Informática para a Musicologia

3.º Ano de Lic.ª em Música + Opção
Universidade do Minho
Ano Lectivo de 2022/23

Exame de Recurso — 31 de Janeiro 2023, 09h00, Sala E7-0.05

---

*Esta prova consta de **8** questões todas com a mesma cotação.*

PROVA INDIVIDUAL SEM CONSULTA (2h)

Grupo I

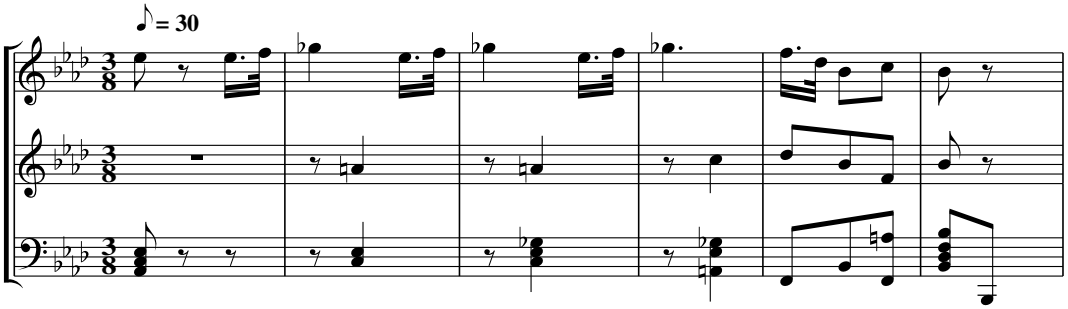
Questão 1 Nesta disciplina, ao longo das aulas, foi utilizada extensivamente a plataforma on-line JUPYTER NOTEBOOK, que é hoje uma referência no desenvolvimento de aplicações informáticas. Um "notebook" em Jupyter é uma sequência de células, que podem ser de três diferentes tipos: *Markdown*, *Code* ou *Raw*.

Estabeleça a diferença entre esses três tipos de células, indicando o seu propósito e a linguagem que é usada em cada caso.

Questão 2 Indique, justificando, qual o resultado de se avaliarem as células seguintes num Jupyter notebook com "kernel" em Haskell:

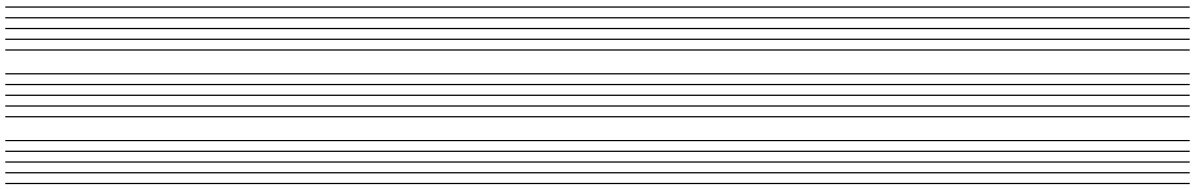
1. $\boxed{(\mathsf{init} \cdot \mathsf{tail} \cdot \mathsf{init})\ [1\,..\,10]}$
2. $\boxed{(\mathit{unwords} \cdot \mathsf{reverse} \cdot \mathit{words})\ \texttt{"JOANA MARIA"}}$
3. $\boxed{(\mathsf{take}\ 5 \cdot \mathit{nub})\ [\texttt{"JOANA"}, \texttt{"MARIA"}]}$
4. $\boxed{\mathit{nrep}\ [(\texttt{"a"}, 1), (\texttt{"b"}, 2), (\texttt{"b"}, 3)]}$

Questão 3 Tomando por base `L:1/8`, escreva em notação ABC o seguinte fragmento de três instrumentos não identificados de uma dada obra:



Questão 4 Transcreva para os pentagramas abaixo o seguinte fragmento de ABC:

```
X:7
M:6/4
L:1/4
K:G
V: 1 clef=treble name="Violino I.mo" sname="Vl.I"
V: 2 clef=treble name="Violino II.do" sname="Vl.II"
V: 3 clef=treble name="Violino III.zo" sname="Vl.III"
%%staves  [1 2 3]
[V: 1] [Q:"Largo"] "_[fol.7 (18)]"  z | Z9 | (=f e/>^f/a/<=g/ f/)>(g/ e e |\
       e/>f/ ^d)  .B (=d/<=c/ B) (d | e2) \
[V: 2] [Q:"Largo"] "_[fol.15 (34)]" z | Z9 | B2 (e2 ^d) (B | \
        =c B/>)(^c/e//d3/4 d c/>)(d/ B | B/>^c/ ^A)
[V: 3] [Q:"Largo"] "_[fol.23 (50)]" z | Z9 | G2 (G F2) (A | \
        A G F G2) (^c | F2)
%--------------------------------
```



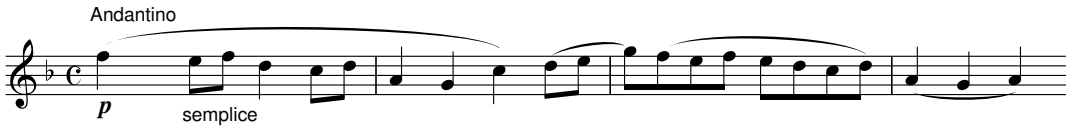
Questão 5 Pretendendo-se programar uma operação $temp$ que extraia o tempo total que uma melodia demora a tocar, complete as seguintes propriedades dessa operação:

$temp\ [\,] = \ldots$
$temp\ [(n, d)] = \ldots$
$temp\ (x +\!\!+\ y) = \ldots$

De seguida, proceda como se fez nas aulas para obter um programa em Haskell que implemente $temp$.

Grupo II

Perto do final da sua vida, Johannes Brahms (1833-1897) escreveu várias obras para clarinete, sendo uma delas o célebre quinteto opus 115, de que se dá a seguir o início da parte de clarinete (em A) de um dos andamentos:



A partitura foi gerada em Jupyter notebook correndo a célula

$\boxed{abcPlayM\ \texttt{"F"}\ \texttt{"C"}\ brahms}$

tendo-se definido, previamente:

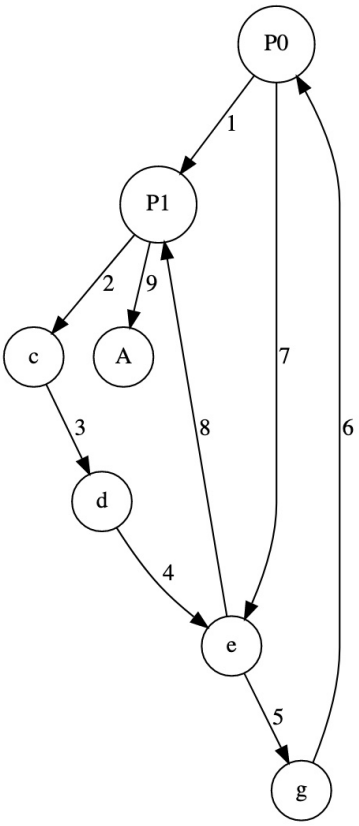
$$
\begin{aligned}
brahms = [&(\texttt{"f"}, 1\,\%\,4), (\texttt{"e"}, 1\,\%\,8), (\texttt{"f"}, 1\,\%\,8), (\texttt{"d"}, 1\,\%\,4), (\texttt{"c"}, 1\,\%\,8), (\texttt{"d"}, 1\,\%\,8),\\
&(\texttt{"A"}, 1\,\%\,4), (\texttt{"G"}, 1\,\%\,4), (\texttt{"c"}, 1\,\%\,4), (\texttt{"d"}, 1\,\%\,8),\\
&(\texttt{"e"}, 1\,\%\,8), (\texttt{"g"}, 1\,\%\,8), (\texttt{"f"}, 1\,\%\,8), (\texttt{"e"}, 1\,\%\,8), (\texttt{"f"}, 1\,\%\,8),\\
&(\texttt{"e"}, 1\,\%\,8), (\texttt{"d"}, 1\,\%\,8), (\texttt{"c"}, 1\,\%\,8), (\texttt{"d"}, 1\,\%\,8), (\texttt{"A"}, 1\,\%\,4),\\
&(\texttt{"G"}, 1\,\%\,4), (\texttt{"A"}, 1\,\%\,4)]
\end{aligned}
$$

Questão 6 Tendo-se executado, em Jupyter, a célula

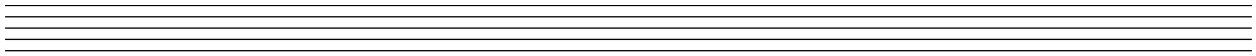
```
(m, r) = unzip brahms
—
reduced m
```

obteve-se o grafo ao lado.

Identifique os "motifs" P_0 e P_1, justificando.



Questão 7 Escreva no pentagrama

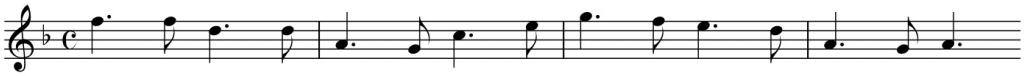


o que se obterá em Jupyter se se mandar executar a célula:

```
(abcPlayM "F" "C" · delay 1/2 · reverse) brahms
```

Justifique a sua resposta.

Questão 8 Caracterize a sequência de amostragem s que produz a seguinte redução do original de Brahms acima transcrito,



quando se executa a célula seguinte:

```
s = ..........
—
(abcPlayM "F" "C" · sample s) brahms
```

Catálogo de operações estudadas e usadas em ambiente Jupiter

Básicas

- succ *(sucessor)* — assumida uma ordem, succ x vai buscar o próximo elemento de x, nessa ordem
- pred *(predecessor)* — inversa da anterior
- $(\equiv)$ *(igualdade)* — $a \equiv b$ diz se a e b são o mesmo objecto
- $f \cdot g$ *(depois de)* — aplicar f depois de g, i.e.: $(f \cdot g)\ a = f\ (g\ a)$
- id *(não fazer nada)* — $id\ x = x$ qualquer que seja x

Sequências

- $(+\!\!+)$ *(junção)* — juntar duas sequências numa só
- $(:)$ *(aposição)* — $a : x$ é a mesma coisa que a junção $[\,a\,] +\!\!+ x$
- drop i *(obter sufixo)* — elimina os primeiros i-elementos da sequência
- filter p *(filtragem)* — filter $p\ m$ vai selecionar os elementos de m que obedecem à condição p
- head *(obter cabeça)* — obter o primeiro elemento de uma sequência, caso exista
- init *(obter iniciais)* — obter todos menos o último elemento de uma sequência, caso existam
- last *(obter último)* — obter o último elemento de uma sequência, caso exista
- length *(comprimento)* — dá o número de letras de uma palavra ou, no caso geral, o comprimento de uma sequência
- $lines$ *(linhas de um texto)* — divide um texto na sequência das suas linhas
- map *(aplica a todos)* — map $f\ s$aplica a operação fa todos os elementos da sequência s, pela ordem com que ocorrem
- nub *(remoção de repetidos)* — eliminar os elementos repetidos de uma sequência
- reverse *(inversão)* — inverter uma sequência
- $sort$ *(ordenação)* — ordenar uma sequência por ordem crescente
- sum *(somatório)* — soma todos os elementos de uma sequência numérica
- tail *(obter cauda)* — obter todos menos o primeiro elemento de uma sequência, caso existam
- take i *(obter prefixo)* — dá os primeiros i-elementos da sequência
- $unlines$ *(inversa da anterior)* — junta uma sequência de linhas num só texto
- $unwords$ *(inversa da anterior)* — junta palavras numa só frase, separadas por espaços
- $words$ *(palavras de uma frase)* — divide uma frase na sequência das suas palavras

Pares

- π_1 *(o primeiro)* — dá o primeiro elemento do par (a, b), isto é a
- π_2 *(o segundo)* — dá o segundo elemento do par (a, b), isto é b
- $f \times g$ *(paralelo)* — aplica f e g em paralelo, i.e. ao mesmo tempo: $(f \times g)\ (a, b) = (f\ a, g\ b)$. Por ser mais prático, usamos $f \times g$ em vez de $f \times g$

Sequências de pares (música)

- $ddrop\ d$ *(obter sufixo por duração)* — $ddrop\ d\ m$ vai buscar as notas que $dtake\ d\ m$ não seleciona
- $dtake\ d$ *(obter prefixo por duração)* — $dtake\ d\ m$ vai buscar tantas notas quanto possível m até prefazer a duração d
- $nrep$ *(ligaduras)* — notas consecutivas com a mesma altura de som são ligadas numa única nota com a duração total correspondente
- $sample$ *(amostragem)* — $sample\ s\ m$ amostra a melodia $m$segundo a sequência de durações s
- unzip *(o contrário de* zip*)* — $(a, b) =$ unzip m significa que em a vamos ter map $\pi_1\ m$ e em b vamos ter map $\pi_2\ m$

Abc

- $abcplease$ *(mostrar partitura)* — $abcplease\ m$ mostra a partitura de m sem armação de clave nem compasso, nem barras de compasso
- $abcPlayM$ *(mostrar partitura)* — $abcPlay\ K\ C\ m$ - mostra a partitura de m com armação de clave K, compasso C e barras de compasso (regular) deduzidas a partir de C
- $abcPlay_$ *(mostrar partitura)* — $abcPlay_\ K\ C\ m$ mostra a partitura de m com armação de clave K e compasso C
- $abcPlay$ *(mostrar partitura)* — $abcPlay\ K\ C\ c\ m$ - mostra a partitura de m com armação de clave K, compasso C e barras de compasso (e.g. irregular) segundo c