



Avaliação 1
Segunda-Chamada
Estrutura de Dados
Professora Tatiane Fernandes



UNIVERSIDADE
FEDERAL DO CEARÁ

Nome:

Matrícula:

- 1 – A interpretação também faz parte da avaliação.
- 2 – Você poderá utilizar caneta (preta ou azul) ou escrever a lápis.
- 3 – Pedidos de revisão de questões feitas à lápis serão apenas aceitas no dia e horário de entrega e correção da avaliação.
- 4 – Em caso de detecção de utilização de qualquer meio fraudulento na concepção das respostas da avaliação, os envolvidos terão suas notas zeradas.

Com a grande repercussão da performance dos cursos de Engenharia de Software e Ciência da Computação (Notas 4 e 5 no MEC)! Várias empresas se interessaram em contratar alunos da UFC - Campus Russas!

Suponha que a empresa Spotify abriu uma vaga para desenvolvedor junior, onde os pré-requisitos são: conhecimento avançado em Linguagem C e Estrutura de Dados. Você fica muito interessado no alto \$alário oferecido pela empresa e decide se inscrever no processo seletivo. Então logo na primeira fase, você recebe o seguinte desafio:

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4 typedef struct No{
5     char * nome_musica;
6     double minutos;
7     struct No * prox;
8     struct No * ant;
9 }No;
10
11 No * inicio = NULL;
12
13 void adicionar(char * nome_musica, double minutos){
14
15     No * novo = malloc(sizeof(No));
16     novo->nome_musica = nome_musica;
17     novo->minutos = minutos;
18     novo->prox = NULL;
19     novo->ant = NULL;
20
21     if(inicio == NULL){ //lista vazia!
22         inicio = novo;
23         novo->prox = novo;
24         novo->ant = novo;
25     }else{ //
26         //você deve codificar
27         //e explicar esta parte
28     }
29 }
```

```
31 void play(No * aux, int num_repeticoes){
32     //você deve codificar e explicar esta parte
33 }
34
35 void remover(){
36     //você deve codificar e explicar esta parte
37 }
38
39 int main(){
40     adicionar("Evoque Prata", 1.47);
41     adicionar("Eu gosto assim", 2.28);
42     adicionar("Pacto", 3.03);
43     adicionar("Bombonzinho", 2.34);
44     remover();
45     printf("Top Spotify Brasil hoje:\n");
46     play(inicio, 2);
47     return 0;
48 }
```

- 1 – (2,5). Codifique e explique como deve ficar a parte do código ausente na função **adicionar**. Lembre-se que a ordem das músicas do Spotify segue, é a ordem que elas são adicionadas na lista!
- 2 - (2,5). Codifique e explique como deve ficar a parte do código ausente na função **remover**. A função **remover** deve sempre que for chamada, remover a música que possui maior tempo (minutos) da lista.
- 3 – (2,5). Codifique e explique como deve ficar a parte do código ausente da função **play**. A função **play** DEVE apresentar todos os nomes de músicas e o seu tempos (minutos) na ordem que estão na lista, esta apresentação deve ser repetida o número de vezes que a variável **num_repeticoes** contiver. **ESTA FUNÇÃO DEVE SER 100% RECURSIVA.**
- 4 - (2,5) (a) Após a inserção dos códigos ausentes e execução do mesmo, apresente como ficaria o resultado "printado" no terminal. (b) Considerando a análise de complexidade simplificada vista em sala de aula (**linear ou constante**). Apresente a complexidade de cada um dos **casos das funções de adicionar e remover** do código acima, justifique de forma curta cada uma delas.