

Que tal criar o seu próprio sistema de **Recomendação de Músicas**?

Na etapa anterior deste desafio você foi capaz de implementar um algoritmo que analisa músicas e artistas de uma coleção de músicas obtidas diretamente do Spotify API. Para isto, você usou seus conhecimentos em programação e em projeto e análise de algoritmos para criar uma solução eficiente que apresenta ao usuário a quantidade de músicas de cada artista em ordem de popularidade.

Sua próxima tarefa envolve recomendar músicas para o usuário considerando o seu gosto musical. Ou seja, dadas Q músicas de entrada e um valor de K para o seu algoritmo, você deve apresentar ao usuário as K músicas mais similares a cada música consultada.

Esta tarefa envolve as seguintes etapas:

- ✓

1) Ler o arquivo CSV;

Já implementado na parte 1
- ✓

2) Armazenar os dados em memória dinamicamente;
- ✓

3) Ordenar os dados lidos do csv em ordem alfabética pelo **id das músicas**; (Basta adaptar o campo de ordenação para track_id)
- 4) Calcular a matriz de dissimilaridade entre todas as músicas;

4.1) Aplicar a distância euclidiana sobre as 9 características apresentadas no dataset;
- 5) Listar as K músicas mais similares às músicas dadas pelo usuário;

5.1) Realizar uma **Busca Binária** nas linhas da matriz já ordenada

5.2) Realizar a ordenação da linha de acordo com as **menores distâncias** usando o algoritmo **Radix Sort**

5.3) Mostrar os **K primeiros valores** do vetor ordenado

Entrada:

A entrada contém (2 + Q) linhas, a primeira sendo o nome do arquivo CSV, a segunda o valor Q e K, seguidas de Q linhas contendo o ID das músicas a serem buscadas por similaridade.

Os dados a serem analisados estão contidos em um arquivo CSV, o qual está formatado da seguinte maneira. (O mesmo formato do trabalho 1)

A primeira linha contém o cabeçalho, apresentando o nome de cada um dos atributos. As demais linhas contém os dados das músicas, sendo que há uma música por linha.

As colunas são separadas por ; e possuem os seguintes atributos: track_name; track_id; album_name; album_id; artist_name; artist_id; release_date; length; popularity; acousticness; danceability; energy; instrumentalness; liveness; loudness; speechiness; tempo; time_signature;

Note que os seis primeiros atributos [1-6] são **nomes** e **identificadores** das músicas, albums e artistas, respectivamente. Os atributos [7-9] são a data de **lançamento da música**, a **duração** em milissegundos e a **popularidade** da música no momento em que o script obteve as informações do Spotify. Por fim, existem 9 atributos [10-18] que são características das músicas, as quais o próprio Spotify oferece para cada música.





Saída

Você deverá imprimir na saída padrão as K músicas mais similares à cada música consultada, conforme a seguinte função ilustra:

```
void print_most_similar_tracks(track_t *tracks, dist_t **M, int n_tracks, int index, int K){
    printf("----As %d musicas mais parecidas com %s sao:\n", K, tracks[index].track_name);
    for (int i=0; i<K; i++){
        printf("\t(%d)Artista: %s\n", i, tracks[M[index][i].index_j].artist_name);
        printf("\t\tMusica: %s\n", tracks[M[index][i].index_j].track_name);
        printf("\t\tDissimilaridade: %lf\n", M[index][i].distance);
        printf("\t\tLink: https://open.spotify.com/track/%s\n", tracks[M[index][i].index_j].track_id);
    }
    printf("\n");
}
```

Exemplo de Entrada

```
1 caso0.csv
2 2 5
3 6hv1qtVzwDUkSY6WW3klcE
4 64Z7z09eYMqznZAwlvXj2i
```

Exemplo de Saída

```
1 ----As 5 musicas mais parecidas com Lôra Ao Vivo sao:
2 (0)Artista: Tiago Iorc
3 Musica: Lôra Ao Vivo
4 Dissimilaridade: 0.000000
5 Link: https://open.spotify.com/track/6hv1qtVzwDUkSY6WW3klcE
6 (1)Artista: Tiago Iorc
7 Musica: Unordinary Gold
8 Dissimilaridade: 2.778703
9 Link: https://open.spotify.com/track/4821zH2mqQifPnDKB7ifAa
10 (2)Artista: Tiago Iorc
11 Musica: A Vida Nunca Cansa
12 Dissimilaridade: 4.446614
13 Link: https://open.spotify.com/track/06oNiLeQ7AKuALj1Vt71Nj
14 (3)Artista: Tiago Iorc
15 Musica: Tangerina Ao Vivo
16 Dissimilaridade: 4.639121
17 Link: https://open.spotify.com/track/6TDPTk5c9r1aAaLEILqNmA
18 (4)Artista: Nenhum De Nós
19 Musica: Dias Que Virão Ao Vivo
20 Dissimilaridade: 5.890278
21 Link: https://open.spotify.com/track/6shuD80IZe8iWofqQNUUD3
22
23 ----As 5 musicas mais parecidas com Hoje Lembrei Do Teu Amor sao:
24 (0)Artista: Tiago Iorc
25 Musica: Hoje Lembrei Do Teu Amor
26 Dissimilaridade: 0.000000
27 Link: https://open.spotify.com/track/64Z7z09eYMqznZAwlvXj2i
28 (1)Artista: Engenheiros Do Hawaii
29 Musica: Quem Tem Pressa Não Se Interessa
30 Dissimilaridade: 0.888939
31 Link: https://open.spotify.com/track/1D7CXJGQldz19dItJVMED
32 (2)Artista: Engenheiros Do Hawaii
33 Musica: Quem Tem Pressa Não Se Interessa
34 Dissimilaridade: 0.888939
35 Link: https://open.spotify.com/track/2n4XzHMVXN2mIkiNdscXfr
36 (3)Artista: Tequila Baby
37 Musica: Tira o Sutiã Tira a Calcinha
38 Dissimilaridade: 0.956972
39 Link: https://open.spotify.com/track/0uF3gW5GKkCIPfqAWqogeK
40 (4)Artista: Tiago Iorc
41 Musica: Proibida Pra Mim
42 Dissimilaridade: 1.062896
43 Link: https://open.spotify.com/track/49qaKxCTfFovGl7A17UOCU
44
```

Cálculos necessários

Arquivo CSV

track_name	track_id	...	f ₁	f ₂	f ₃	f ₄	f ₅	f ₆	f ₇	f ₈	f ₉
musica_1	tID_musica_1	...	f ₁	f ₂	f ₃	f ₄	f ₅	f ₆	f ₇	f ₈	f ₉
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
musica_N	tID_musica_N	...	f ₁	f ₂	f ₃	f ₄	f ₅	f ₆	f ₇	f ₈	f ₉

acousticness;
danceability;
energy;
instrumentalness;
liveness;
loudness;
speechiness;
tempo;
time_signature;

Matriz de Dissimilaridade

		Músicas					
		0	1	2	3	4	5
Músicas	0	d	d	d	d	d	d
	1	d	d	d	d	d	d
	2	d	d	d	d	d	d
	3	d	d	d	d	d	d
	4	d	d	d	d	d	d
	5	d	d	d	d	d	d

Observações

A matriz de dissimilaridade M à esquerda é Quadrada e Simétrica.

A **Diagonal principal** possui o valor d = 0, pois a distância de uma música para ela mesma é 0.

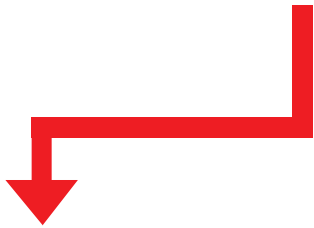
Cada posição M[i][j] da matriz é a distância da música [i] para a música [j] ([Todas com Todas](#))

A distância euclidiana d é calculada utilizando as 9 características das músicas. ([Veja a seguir](#))

Dissimilaridade entre Músicas

		Músicas			
		0	1	2	3
Músicas	0	0	d	d	d
	1	d	0	d	d
	2	d	d	0	d
	3	d	d	d	0

$d = \text{distancia}(\text{musica}[0], \text{musica}[1]);$
 $=$
 $d = \text{distancia}(\text{musica}[1], \text{musica}[0]);$



$$d = \sqrt{(m_2f_1 - m_1f_1)^2 + \dots + (m_2f_9 - m_1f_9)^2}$$

