1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "music linkedlist.h"
Musicas initMusicas(){
        int i;
        Musicas aux = (Musicas)malloc(sizeof(Musica));
        aux->title = NULL;
        aux->author lyric = NULL;
        aux->author music = NULL;
        aux->author = NULL;
        aux->from = NULL;
        aux->singer = NULL;
        aux->estrofe=NULL;
        aux->letra = NULL;
        aux->next = NULL;
        return aux;
}
Letra initLetra(){
        return NULL;
}
void freeList(Letra l){
    if(l){
        if(l->next != NULL)
            freeList(l->next);
        free(l);
    }
}
void freeMusica(Musicas m){
    if(m){}
        if(m->next != NULL)
            freeMusica(m->next);
        free(m);
    }
}
Letra copyLetra(Letra l)
{
    Letra new=initLetra();
    Letra aux = l;
    while(aux != NULL)
        new = insertLetra(new, aux->line);
        aux = aux->next;
    return new;
}
** ADD'S
/*Adicionar musica a lista de musicas */
Musicas insertMusic(Musicas m, Musicas new){
        if(m == NULL){
                m = new;
                m->next = NULL;
        else{
```

```
new->next = m;
                m = new;
        return m;
/*Adicionar um verso a letra*/
Letra insertLetra(Letra l, char* v){
        if(l == NULL){
                l = (Letra)malloc(sizeof(letras));
                l->line = strdup(v);
                l->next = NULL;
        else{
                Letra aux = NULL;
                aux = l;
                while(aux->next != NULL){
                        aux = aux->next;
                Letra new = (Letra)malloc(sizeof(letras));
                new->line = strdup(v);
                new->next = NULL;
                aux->next = new;
        return l;
   Set's para ir contruindo uma musica *
void setTitle(Musicas m, char *t){
        while(*t==' ') t++;
        m->title = strdup(t);
void setLyric(Musicas m, char *l){
        m->author_lyric = strdup(l);
void setAutMusic(Musicas m, char *a){
        m->author music = strdup(a);
}
void setAuthor(Musicas m, char *a){
        m->author = strdup(a);
void setFrom(Musicas m, char *f){
        m->from = strdup(f);
void setSinger(Musicas m, char *s){
        m->singer = strdup(s);
}
void setLetra(Musicas m, Letra l){
        m->letra = (Letra)malloc(sizeof(letras));
        m->letra = l;
}
** Array de inteiros que cada indice representa uma estrofe, em
  que representa o numero de vertices que a contituem
/*i - nº da estrofe
* n - nº de versos
void setEstrofe(Musicas m, int i ,int n){
        m->estrofe = realloc(m->estrofe, sizeof(int)*(i+1));
```

3

```
m->estrofe[i]=n;
}
** GET'S
Musicas getMusic(Musicas m, int indice){
        Musicas res = (Musicas)malloc(sizeof(Musica));
        Musicas aux = (Musicas)malloc(sizeof(Musica));
        aux = m;
        int flag = 0;
        int j = 1;
        while(flag == 0){
                if(aux == NULL){
                        //out = 1;
                         return NULL;
                }
                else if(j == indice){
                         res = aux;
                         res->next = NULL;
                         flag = 1;
                }
                else{
                        aux = aux->next;
                         j++;
                }
        return res;
}
** LENGHT
int lenghtMusicas(Musicas m){
 Musicas aux = m;
  int t = 0;
  while(aux != NULL){
    t++;
    aux = aux->next;
  return t;
}
int lenghtLetra(Letra m){
 Letra aux = m;
  int t = 0;
  while(aux != NULL){
    t++;
    aux = aux->next;
  return t;
```