```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>

#include "makeLatex.h"



#define BEGIN_DOCUMENT "\\begin{document}\n"
#define END_DOCUMENT   "\\end{document}\n"
#define MAKETITLE      "\\maketitle\n"
#define NEW_LINE       "\\newline"



Map iniateMapLatex() {
  Map p = (Map) malloc(sizeof(Map));

  p->count=0;

  p = insertRule(p, "*" , "" );
  p = insertRule(p, "«" , "$\\ll$" );
  p = insertRule(p, "»" , "$\\gg$" );
  p = insertRule(p, "_" , "\\_" );
  p = insertRule(p, "#" , "\\#" );
  p = insertRule(p, "´" , "'" );
  p = insertRule(p, "[" , "(" );
  p = insertRule(p, "]" , ")" );
  p = insertRule(p, "-" , "" );

  return p;
}

Map insertRule(Map p, char* simb, char* expr) {
  if(p==NULL) {
    p = (Map) malloc(sizeof(Map));
    p->count=0;
  }
  (p->count)++;
  p->simb = (char**) realloc(p->simb,sizeof(char*)*(p->count));
  p->value = (char**) realloc(p->value,sizeof(char*)*(p->count));

  p->simb[p->count-1]=simb;
  p->value[p->count-1]=expr;

  return p;
}


char* filterScharacters(char * src, Map mapa) {

  char* aux = malloc(strlen(src)+1);
  memset(aux,'\0',strlen(src)+1);

  int n=0, i=0;
  while (*src!='\0') {
    int flag=0;
    for(i=0; i< mapa->count;i++) {

      if(strncmp(src,mapa->simb[i],strlen(mapa->simb[i]))==0) {
        int size=0;
        if(strlen(mapa->value[i])==0) {
          src+=strlen(mapa->simb[i])-1;
```

```
          n--;
          flag=1;
        } else {
          size=(strlen(src)+strlen(aux)+strlen(mapa->value[i])-strlen(mapa->simb[i])+1);

          //criação de buffer
          char* temp = NULL;
          temp=realloc(temp,size);
          memset(temp,'\0',size);

          //copia existente e valor novo
          strcpy(temp,aux);
          strcat(temp,mapa->value[i]);
          aux=temp;

          //incrementa posições
          n+=strlen(mapa->value[i])-1;
          src+=strlen(mapa->simb[i])-1;

          flag=1;
        }
      }
    }
    if(flag==0) aux[n]=*src;

    src++;n++;
  }
  return aux;
}




char * createTitle(char* title)
{
    char open[] = "\\title{";
    char close[] = "}";
    char * latextitle = malloc(strlen(title) + strlen(open) + strlen(close)+1);
    latextitle = strcpy(latextitle,open);
    latextitle = strcat(latextitle,title);
    latextitle = strcat(latextitle,close);
    return latextitle;
}


char * createAuthor1(char* author)
{
    char opentag[] = "\\author{";
    char closetag[] = "}";
    char* latexauthor = malloc(strlen(author) + strlen(opentag) + strlen(closetag)+1);

    latexauthor = strcpy(latexauthor,opentag);
    latexauthor = strcat(latexauthor,author);
    latexauthor = strcat(latexauthor,closetag);
  return latexauthor;
}

char * createAuthor2(char* authorL, char* authorM)
{

    char sep[] = "\\\\";

    char opentag[] = "\\author{";
    char closetag[] = "}";
    char* latexauthor = malloc(strlen(authorL) + strlen(authorL)+ strlen(opentag) + strlen(sep)
+strlen(closetag)+1);
```

```
    latexauthor = strcpy(latexauthor,opentag);
    latexauthor = strcat(latexauthor,authorL);
    latexauthor = strcat(latexauthor,sep);
    latexauthor = strcat(latexauthor,authorM);
    latexauthor = strcat(latexauthor,closetag);
    return latexauthor;
}

char* listFrase(char* elem)
{
    char brk[] = "\\newline";
    char* listFrase = malloc(strlen(elem) + strlen(brk)+1);
    listFrase = strcpy(listFrase,elem);
    listFrase = strcat(listFrase,brk);
    return listFrase;
}

char * createSinger(char* singer)
{
    char opentag[] = "\\begin{flushright}";
    char closetag[] = "\\end{flushright}";
    char* latexsinger = malloc(strlen(singer) + strlen(opentag) + strlen(closetag)+1);

    latexsinger = strcpy(latexsinger,opentag);
    latexsinger = strcat(latexsinger,singer);
    latexsinger = strcat(latexsinger,closetag);
    return latexsinger;
}


char * initLatex()
{
  char init[] = "\\documentclass{article}\n\\usepackage[utf8]{inputenc}\n\\date{}\n\\raggedright\n";
  char* res = malloc(strlen(init)+1);
  res = strcpy(res,init);
  return res;
}

char * closeLatex()
{
  char close[] ="\\end{document}\n";
  char* res = malloc(strlen(close)+1);
  res = strcpy(res,close);
  return res;
}


void printMusica(Musicas music)
{
    printf("%s\n",createTitle(music->title));

    if(music->author_lyric != NULL && music->author!= NULL)
        printf("%s\n",createAuthor2(music->author_lyric,music->author));
    else{
        if(music->author_lyric != NULL)
            printf("%s\n",createAuthor1(music->author_lyric));
          else{
                printf("%s\n",createAuthor1(music->author));
          }
    }

    printf("%s\n",BEGIN_DOCUMENT);
    printf("%s\n",MAKETITLE);

    if(music->letra != NULL){
        Letra letra = music->letra;
        int i = 0;
```

```
        int v = 0;
        while(letra != NULL){
            v = music->estrofe[i];
          //  printf("%s\n",BEGIN_CENTER);
            while(v>0){
              printf("%s\n",listFrase(letra->line));
              v--;
              letra=letra->next;
            }
            //printf("%s\n",END_CENTER);
            i++;
            printf("%s\n",NEW_LINE);
        }
    }

    if(music->singer != NULL){
      printf("%s\n",createSinger(music->singer));
    }
}


void convertTex2Pdf (char *nome){
    char program[] = "pdflatex ";
    int size = strlen(program)+strlen(nome)+2+13;

    char * command = malloc(size);
    memset(command,'\0',size);


    strcpy(command,program);
    strcat(command,"\"");
    strcat(command,nome);
    strcat(command,"\"");
    strcat(command," 1>/dev/null");


    int i;
    i=system(command);
    i=system("rm *.log");
    i=system("rm *.aux");

    free(command);
}


void saveMusica(Musicas music)
{


  FILE*f = NULL;

  int size;
  char* nome = NULL;
  if  ((music->author != NULL)) {
    size = strlen(music->author)+strlen(music->title)+5+3;
    nome = malloc(size);
    strcpy(nome,music->author);
    strcat(nome," - ");
    strcat(nome, music->title);
  } else {
    size = strlen(music->title)+5;
    nome = malloc(size);
    strcpy(nome,music->title);
  }

  strcat(nome,".tex");
  f = fopen(nome, "w");
```

```
  if (f != NULL){
    fprintf(f,"%s\n",initLatex());
    fprintf(f,"%s\n",createTitle(music->title));

    if(music->author_lyric != NULL && music->author!= NULL)
        fprintf(f,"%s\n",createAuthor2(music->author_lyric,music->author));
    else{
        if(music->author_lyric != NULL)
            fprintf(f,"%s\n",createAuthor1(music->author_lyric));
          else{
                fprintf(f,"%s\n",createAuthor1(music->author));
          }
    }
    fprintf(f,"%s\n",BEGIN_DOCUMENT);
    fprintf(f,"%s\n",MAKETITLE);

    if(music->letra != NULL){
        Letra letra = music->letra;
        int i = 0;
        int v = 0;

        while(letra != NULL){
            v = music->estrofe[i];
            while(v>0){
              fprintf(f,"%s\n",listFrase(letra->line));
              v--;
              letra=letra->next;
            }
            i++;
            fprintf(f,"%s\n",NEW_LINE);
        }
    }




    if(music->singer != NULL){
      fprintf(f,"%s\n",createSinger(music->singer));
    }
    fprintf(f,"%s\n",END_DOCUMENT);
    fclose(f);
  }
  convertTex2Pdf(nome);
}


void makeLatex(Musicas musicas)
{
  char dir[] = "converted";
  int res;

  struct stat dirStat;
  if((res=stat(dir, &dirStat))<0)
      mkdir(dir, 0700);



  if(chdir(dir)!=0)
    printf("ERRO - Ocorreu erro ao mudar de directoria, não foi possivel efectuar conversão!\n");
  else {
    if(musicas->author != NULL)
      printf("Done! - %s - %s\n",musicas->author,musicas->title);
    else
      printf("Done! - %s\n",musicas->title);
    while(musicas != NULL)
    {
```

```
      saveMusica(musicas);
      musicas = musicas->next;
    }

  }
  if(chdir("..")!=0)
    printf("ERRO - Ocorreu erro ao mudar de directoria, verifique se a conversão foi efectuada!\n");

}
```