```
import UIKit
class AddSongViewController: UIViewController {
    @IBOutlet weak var songInput: UITextField!
    @IBOutlet weak var artistInput: UITextField!
    @IBOutlet weak var lyricsInput: UITextView!
    var importedTab = ""
    var importedTitle = ""
    var importedArtist = ""
    //add button clicked
    @IBAction func addSongButton(_ sender: UIButton) {
        //when button clicked save title and artist into array
        if (songInput.text != ""){
            songTitle.append(songInput.text!)
            artist.append(artistInput.text!)
            let defaults = UserDefaults.standard
            defaults.set(songTitle, forKey: "SongList")
            let defaultsArtist = UserDefaults.standard
            defaultsArtist.set(artist, forKey: "Artist")
            }
        //when button pressed write on file or create file
        let tabFile = songInput.text!
        let DocumentDirURL = try! FileManager.default.url(for: .
            documentDirectory, in: .userDomainMask, appropriateFor: nil,
            create: true)
        let fileURL = DocumentDirURL.appendingPathComponent(tabFile).
            appendingPathExtension("txt")
        let writeString = lyricsInput.text!
        do{
            try writeString.write(to: fileURL, atomically: true, encoding:
                String.Encoding.utf8)
        } catch let error as NSError {
            print("Error: \(error)")
        }
        //reset fields
        songInput.text = ""
        artistInput.text = ""
        lyricsInput.text = ""
        //removing first responder status
        self.songInput.resignFirstResponder()
        self.artistInput.resignFirstResponder()
        self.lyricsInput.resignFirstResponder()
        _ = navigationController?.popToRootViewController(animated: true)
    }
    //Give up first responder at touch screen
    override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
        self.view.endEditing(true)
```

```
}
override func viewDidLoad() {
    super.viewDidLoad()
    //setting up text view border temporarily
    lyricsInput.layer.borderWidth = 1
    self.lyricsInput.layer.borderColor = UIColor(red: 214/255, green: 214/
        255, blue: 214/255, alpha: 1).cgColor
    if importedTab != ""{
        lyricsInput.text = importedTab
        songInput.text = importedTitle
        artistInput.text = importedArtist
        lyricsInput.font = UIFont(name: "CourierNewPSMT", size: 18)
        songInput.font = UIFont(name: "CourierNewPSMT", size: 18)
        artistInput.font = UIFont(name: "CourierNewPSMT", size: 18)
    }
}
//covert lyrics when convert button is pressed
@IBAction func convertButton( sender: UIButton) {
    let fileName = "chordsList"
    let DocDirURL = try! FileManager.default.url(for: .documentDirectory,
        in: .userDomainMask, appropriateFor: nil, create: true)
    let fileURL = DocDirURL.appendingPathComponent(fileName).
        appendingPathExtension("txt")
    var chordsString = ""
    do{
        chordsString = try String(contentsOf: fileURL)
        var arrayOfChords = chordsString.components(separatedBy: ":")
        var Chords = [[String]]()
        for f in 0..<arrayOfChords.count{</pre>
            arrayOfChords[f] = arrayOfChords[f].replacingOccurrences(of:
                "[", with: "")
            arrayOfChords[f] = arrayOfChords[f].replacingOccurrences(of:
                "]", with: "")
            let tempArray = arrayOfChords[f].components(separatedBy: ",")
            Chords.append(tempArray)
        }
        var StringArray = lyricsInput.text.components(separatedBy: "\n")
        var arravLine = [""]
        //get each cell of the lyrics that was divided by \n and separate
            it by spaces
        for i in 0..<StringArray.count{</pre>
            arrayLine = StringArray[i].components(separatedBy: " ")
            //print(arrayLine)
            //get each new array and compare it to the chords
            var count = 0
            for j in 0..<arrayLine.count{</pre>
                //eliminate empty cells
                if arrayLine[j] != ""{
```

```
//go through each chord to check if it matches the
                             first letter of the arrayLine cell
                         //get first character of cell to compare it with the
                         let start = arrayLine[j].index(arrayLine[j].startIndex,
                             offsetBv: 0)
                         let end = arrayLine[j].index(start, offsetBy: 1)
                         let range = start..<end</pre>
                         let firstChar = arrayLine[j].substring(with: range)
                         //go through each array in chords and check the first
                             cell to compare it with the first character of the
                             string
                         //and find out the chord group
                         for x in 0...<Chords.count{</pre>
                             //check to see if first character of string matches
                             if firstChar == Chords[x][0]{
                                 count = x
                                 break
                             }
                         }
                         //after finding out the chord group, check the chords
                         for eachChord in Chords[count]{
                             if arrayLine[j] == eachChord{
                                 arrayLine[j] = arrayLine[j].
                                     replacingOccurrences(of: arrayLine[j],
                                     with: "<" + arrayLine[j] + ">")
                             }
                         }
                     }
                    StringArray[i] = ""
                     if arrayLine.count - 1 == j{
                         StringArray[i] = arrayLine.joined(separator: " ")
                     }
                }
            }
            for f in 0..<StringArray.count{</pre>
                StringArray[f] = StringArray[f].appending("\n")
            lyricsInput.text = ""
            for temp in StringArray{
                lyricsInput.text = lyricsInput.text.appending(temp + "")
            }
        }catch let error as NSError{
            print("Error: \(error)")
        }
    }
}
```