

```
1 cls
2
3 $lyrics_string=("Quisiera:Ayer:cambiarle:conocí:el:un:final" +
4     ":cielo:al:sin:cuento|Las:sol|Y:barras:un:y" +
5     ":hombre:los:sin:tragos:suelo|Un:han:santo:" +
6     "sido:en:testigo|De:prision|Y:el:una:dolor:" +
7     "canción:que:triste:me:sin:causaste:dueño|Y:" +
8     "y:conocí:to':tus:lo:ojos:que:negros|Y:hiciste" +
9     ":ahora:conmigo|Un:sí:infeliz:que:en:no:el:" +
10    "puedo:amor,:vivir:que:sin:aun:ellos:no:yo|" +
11    "Le:te:pido:supera|Que:al:ahora:cielo:camina" +
12    ":solo:solo:un:sin:deseo|Que:nadie:en:por:tus" +
13    ":todas:ojos:las:yo:aceras|Preguntándole:pueda" +
14    ":a:vivir|He:Dios:recorrido:si:el:en:mundo:verdad" +
15    ":entero|te:el:vengo:amor:a:existe|:decir|")
16
17
18 $song1 = @();
19 $song2 = @();
20 $song3 = @();
21 $lyrics = @();
22 $song1_print = "";
23 $song2_print = "";
24 $duplicates = @();
25
26 #Making the array from the string
27 $lyrics += $lyrics_string.Split(":")
28
29 #One loop to split this into two strings
30 for($i=0;$i -lt $lyrics.Length; $i += 2)
31 {
32     $song1 += $lyrics[$i]
33     $song2 += $lyrics[$i+1]
34 }
35
36 #Build the song lyrics to print for the first song
37 foreach($i in $song1)
38 {
39     $song1_print += $i + " "
40     #write-host ($song1_print)
41 }
42
43 #Build the song lyrics to print for the second song
44 foreach($i in $song2)
45 {
46     $song2_print += $i + " "
47     #write-host ($song1_print)
48 }
49
```

```
50 #Break the lines up for both songs
51 $song1_print = $song1_print -Replace "\|", "`r`n"
52 $song2_print = $song2_print -Replace "\|", "`r`n"
53
54 #Print the lyrics for both songs
55 write-host($song1_print)
56 write-host($song2_print)
57
58 #Check for duplicate words
59
60 #Untangle strings in song1 with |
61 $song1_lower = @()
62 $tangled_pair = @()
63 foreach($i in $song1){
64     if ($i.Contains("|"))
65     {
66         $tangled_pair = $i.Split("|")
67         if($tangled_pair[0].length -ne 0)
68         {
69             $song1_lower += $tangled_pair[0].ToLower();
70         }
71         if($tangled_pair[1].length -ne 0)
72         {
73             $song1_lower += $tangled_pair[1].ToLower();
74         }
75     }
76     else
77     {
78         $song1_lower += $i
79     }
80 }
81
82 #Untangle strings in song2 with |
83 $song2_lower = @()
84 $tangled_pair = @()
85 foreach($i in $song2){
86     if ($i.Contains("|"))
87     {
88         $tangled_pair = $i.Split("|")
89         if($tangled_pair[0].length -ne 0)
90         {
91             $song2_lower += $tangled_pair[0].ToLower();
92         }
93         if($tangled_pair[1].length -ne 0)
94         {
95             $song2_lower += $tangled_pair[1].ToLower();
96         }
97     }
98     else
```

```
99     {
100         $song2_lower += $i
101     }
102 }
103
104 #Collect the duplicates into song3
105 $song3 = @()
106 foreach($j in $song1_lower)
107 {
108     if($song2_lower.Contains($j))
109     {
110         if(-Not $song3.Contains(($j.ToLower())))
111         {
112             $song3 += $j.ToLower()
113         }
114     }
115 }
116 $song3 = $song3 | sort
117
118 write-host("Words that are in both songs:`r`n")
119 write-host $song3
120
121 Read-Host -Prompt "`r`nPress any key to exit"
122
123
124
125
126
127
```