```
 1  /*##############################################################################*/
 2  /*HW06_Furkan_Erdol_131044065_part2.c                                           */
 3  /*_______________________________                                               */
 4  /*Written by Furkan Erdol on April 5, 2015                                      */
 5  /*Description                                                                   */
 6  /*_____________                                                                 */
 7  /*                                                                              */
 8  /*                                                                              */
 9  /*<<<<This program checking the Major and Minor Vowel Harmony according to    */
10  /*rules of the vowel harmony and making a noun Plural>>>>                       */
11  /*
    */
12  /*                                                                              */
13  /*Inputs:                                                                       */
14  /*  -Vowels                                                                     */
15  /*  -Nouns                                                                      */
16  /*Outputs:                                                                      */
17  /*  -Plural nouns                                                               */
18  /*  -Vowel harmony results                                                      */
19  /*..............................................................................*/
20  /*                               Includes                                       */
21  /*..............................................................................*/
22  #include <stdio.h>
23  #include <string.h>
24  #define VOWELS "Vowels.txt"
25  #define NOUNS "Nouns.txt"
26  #define PLURAL "Plural.txt"
27
28  typedef enum
29   {HARD, SOFT, CONSONANT1}
30   major_type;
31
32  typedef enum
33   {FLAT, ROUND, CONSONANT2}
34   minor_type;
35
36  typedef enum
37   {FALSE, TRUE}
38   bool;
39
40  /*Function prototypes*/
41  bool is_major_vh_word(const char* word, const char* v_hard, const char* v_soft);
42  major_type major(const char ch1, const char* v_hard, const char* v_soft);
43  bool is_minor_vh_word( const char* word, const char* v_flat, const char* v_round);
44  minor_type minor(const char ch1, const char* v_flat, const char* v_round);
45  major_type find_last_type(const char* word, const char* v_hard, const char* v_soft);
46  char* make_plural(const char* noun , char* plural_noun, const char* v_hard, const char* v_soft);
47
48
49  int
50  main(void)
51  {
52
53      FILE *fp, *fp2, *fp3; /*File pointers*/
54      char hard[7], soft[7], flat[7], round[7]; /*Input vowels*/
55      char plural[15][20]; /*Output plural nouns*/
56      char word[15][20]; /*Input nouns*/
57      bool control; /*Whether it is harmonies*/
58      int i; /*For loops*/
59      int size=0; /*Number of nouns*/
60
61      /*Open Vowels.txt if couldn't open print the screen warning message*/
62      fp=fopen(VOWELS, "r");
63      if(fp==NULL)
64          printf("Vowels.txt counldn't open...");
65
66      /*Reads vowels from file*/
67      fgets(hard,7,fp);
68      fgets(soft,7,fp);
69      fgets(flat,7,fp);
70      fgets(round,7,fp);
71
```

```
 72      fclose(fp);
 73
 74      /*Open Nouns.txt if couldn't open print the screen warning message*/
 75      fp2=fopen(NOUNS, "r");
 76      if(fp2==NULL)
 77          printf("Nouns.txt counldn't open...");
 78
 79      /*Reads nouns from file*/
 80      i=0;
 81      while(fscanf(fp, " %s", word[i])!=EOF)
 82      {
 83          size++;
 84          i++;
 85      }
 86
 87      fclose(fp2);
 88
 89  /*#########################Print the screen results##########################*/
 90      printf("\n%10cMajor%4cMinor", ' ', ' ');
 91
 92      for(i=0;i<size;i++)
 93      {
 94          printf("\n%-12s", word[i]);
 95
 96
 97          control=is_major_vh_word(word[i], hard, soft);
 98          if(control==1)
 99              printf("T%8c", ' ');
100          else
101              printf("F%8c", ' ');
102
103          control=is_minor_vh_word(word[i], flat, round);
104          if(control==1)
105              printf("T");
106          else
107              printf("F");
108
109      }
110
111      printf("\n\n<<<<Plural of the nouns>>>>");
112      for(i=0;i<size;i++)
113      {
114          make_plural(word[i] , plural[i], hard, soft);
115          printf("\n%s---%s", word[i],plural[i]);
116      }
117
118      printf("\n\n");
119
120      /*Open Plural.txt if couldn't open print the screen warning message*/
121      fp3=fopen(PLURAL, "w");
122      if(fp3==NULL)
123          printf("Plural.txt counldn't open...");
124
125      /*Writes to file plural nouns*/
126      for(i=0;i<size;i++)
127      {
128          make_plural(word[i] , plural[i], hard, soft);
129          fprintf(fp3,"%s\n",plural[i]);
130      }
131
132
133
134      return 0;
135  }
136
137  /*Checks whether the word satisfies the major vowel harmony or not and returns*
138   *TRUE or FALSE                                                              */
139  bool is_major_vh_word(const char* word, const char* v_hard, const char* v_soft)
140  {
141      int hard=0, soft=0; /*Checks whether*/
142      int i, j; /*For loops*/
143      bool major=FALSE;
```

```
144
145         for(i=0;i<strlen(word);i++)
146             for(j=0;j<strlen(v_hard);j++)
147                 if(word[i]==v_hard[j])
148                     hard=1;
149                 else if(word[i]==v_soft[j])
150                     soft=1;
151
152         if((hard==1&&soft==0)||(hard==0&&soft==1))
153             major=TRUE;
154
155
156     return major;
157 }
158
159 /*Takes one character and two lists of hard and soft vowels and checks whether*
160  *the character is a soft vowel or a hard vowel, returns HARD, SOFT           *
161  *or CONSONANT                                                                */
162 major_type major(const char ch1, const char* v_hard, const char* v_soft)
163 {
164     int i; /*For loops*/
165     major_type type=CONSONANT1;
166
167     for(i=0;i<strlen(v_hard);i++)
168         if(ch1==v_hard[i])
169             type=HARD;
170         else if(ch1==v_soft[i])
171             type=SOFT;
172
173     return type;
174 }
175
176 /*Checks whether the word satisfies the minor vowel harmony or not and returns*
177  *TRUE or FALSE                                                               */
178 bool is_minor_vh_word( const char* word, const char* v_flat, const char* v_round)
179 {
180     int flat=0, round=0, testing=0; /*Checks whether*/
181     int i, j; /*For loops*/
182     int control1=0, control2=0, control3=0; /*Finds index of vowels*/
183     bool minor=TRUE;
184     char test[3]="oi", test2[3]="o"; /*Test strings*/
185
186     for(i=0;i<strlen(word);i++)
187     {
188         for(j=0;j<strlen(v_flat);j++)
189         {
190             if(word[i]==v_flat[j])
191             {
192                 flat=1;
193                 control1=i;
194             }
195             else if(word[i]==v_round[j])
196             {
197                 round=1;
198                 control2=i;
199             }
200             else if(word[i]==test[j])
201             {
202                 testing=1;
203                 control3=i;
204             }
205
206             if(word[i]==test2[j]&&i>1)
207                 minor=FALSE;
208          }
209
210     if(flat==1&&control2>control1||round==1&&control3>control2)
211         minor=FALSE;
212
213     }
214
215     return minor;
```

```
216 }
217
218 /*Takes one character and two lists of flat and round vowels and checks       *
219  *whether the character is a soft vowel or a hard vowel, returns FLAT, ROUND  *
220  *or CONSONANT                                                                */
221 minor_type minor(const char ch1, const char* v_flat, const char* v_round)
222 {
223     int i; /*For loops*/
224     major_type type=CONSONANT2;
225
226     for(i=0;i<strlen(v_flat);i++)
227         if(ch1==v_flat[i])
228             type=FLAT;
229         else if(ch1==v_round[i])
230             type=ROUND;
231
232     return type;
233 }
234
235 /*Returns the major type (HARD or SOFT) of the last vowel                     */
236 major_type find_last_type(const char* word, const char* v_hard, const char* v_soft)
237 {
238     int i; /*For loops*/
239     major_type last_type, type;
240
241     for(i=0;i<strlen(word);i++)
242     {
243         type=major(word[i], v_hard, v_soft);
244         if(type==HARD||type==SOFT)
245             last_type=type;
246     }
247
248     return last_type;
249 }
250
251 /*Takes a string “noun” and returns itsplural form in “plural_noun”           *
252  *(an output argument)                                                        */
253 char* make_plural(const char* noun , char* plural_noun, const char* v_hard, const char* v_soft)
254 {
255     char hard_plural[5]="lar", soft_plural[5]="ler"; /*For making plural*/
256     major_type type;
257
258     /*Copies to noun*/
259     strcpy(plural_noun, noun);
260
261     /*Calls find last type function*/
262     type=find_last_type(plural_noun, v_hard, v_soft);
263
264     if(type==HARD)
265         strcat(plural_noun, hard_plural);
266     else if(type==SOFT)
267         strcat(plural_noun, soft_plural);
268
269     return plural_noun;
270 }
271
272 /*############################################################################*/
273 /*                End of HW06_Furkan_Erdol_131044065_part2.c                  */
274 /*############################################################################*/
```