```c
/*************************************************************************/
/*                                                                       */
/*    A sample solution for Homework-2 Part-2.                           */
/*                                                                       */
/*                                                                       */
/*************************************************************************/

/* Include libraries */
#include <stdio.h>


#define EPSILON 0.0001          /* A very small error range
                                   or comparing real numbers*/

#define MAX_NUMBER_OF_ITERS 5   /* maximum number of iterations */

#define NOT_IN_ALPHABET -1      /* functions will return NOT_IN_ALPHABET
                                   if a given char is not in English alphabet */

/* Function prototypes */
/* Get the preferred method id from the user */
int preferredMethod(void);

/* Groups characters by using if clauses */
double ifClauses(char ch);

/* Groups characters by using switch clauses */
double switchClauses(char ch);

/* Groups characters by using user defined functions */
double userDefinedFunctions(char ch);

/*  Checks if a given char is group 0 or not */
int isGroup0(char ch);

/*  Checks if a given char is group 1 or not */
int isGroup1(char ch);

/*  Checks if a given char is group 2 or not */
int isGroup2(char ch);

/*  Checks if a given char is group 3 or not */
int isGroup3(char ch);



int main()
{
    /* Variables */
    char ch;
    double group;
    int preferredMethodID;
    int numberOfIterations = 1;

    /* get the preferred method from user */
    preferredMethodID = preferredMethod();
```

```c

    /* Program will run at most MAX_NUMBER_OF_ITERS iterations */
    while (numberOfIterations <= MAX_NUMBER_OF_ITERS)
    {
        /* Get the char */
        printf("Enter the character (0 to terminate)(%d / %d): ",
                              numberOfIterations, MAX_NUMBER_OF_ITERS);

        scanf(" %c",&ch);   /* There is a space character before %c. Why? */

        if (ch == '0')  {    /* Terminate if user enters 0 */
            printf("Program is terminated by the user.\n");
            return 0;
        }

        /* Run the preferred method */
        switch (preferredMethodID) {
            case 1:
                group = ifClauses(ch);
                break;           /* Forgetting "break" here is a logical error */
            case 2:
                group = switchClauses(ch);
                break;
            case 3:
                group = userDefinedFunctions(ch);
                break;
        }

        /* Check if the given character is in the alphabet */
        if (group >= (NOT_IN_ALPHABET - EPSILON)
                  && group <= (NOT_IN_ALPHABET + EPSILON))
            printf("%c is not in English alphabet.\n", ch);
        else
            printf("%c is in Group-%3.1f.\n", ch, group);


        numberOfIterations++;    /* Do not forget to change loop controller! */
    }

    return 0;
}

int preferredMethod()
{
    int preferredMethodID;

    /* get the preferred method from user */
    printf("Which implementation method do you prefer\n\t");
    printf("(1: If Clauses, 2:Switch Clauses, 3:User-defined Functions):");
    scanf("%d", &preferredMethodID);

    /* if user enters an invalid id, ask for a valid one */
    while (preferredMethodID<1 || preferredMethodID>3)
    {
        printf("Illegal choice!\n");
        printf("Which implementation method do you prefer\n\t");
```

```
113         printf("(1: If Clauses, 2:Switch Clauses, 3:User-defined Functions):");
114         scanf("%d", &preferredMethodID);
115     }
116 
117     return preferredMethodID;
118 }
119 
120  /* Groups characters by using if clauses */
121 double ifClauses(char ch)
122 {
123      /* will return -1 if the given char is not in the alphabet*/
124     double group = NOT_IN_ALPHABET;
125 
126     if ((ch>='a' && ch<='c') || (ch>='A' && ch<='c'))
127         group = 0.0;
128     else if ((ch>='d' && ch<='e') || (ch>='D' && ch<='E'))
129         group = 0.1;
130     else if ((ch>='f' && ch<='g') || (ch>='F' && ch<='G'))
131         group = 0.2;
132     else if ((ch>='h' && ch<='i') || (ch>='A' && ch<='I'))
133         group = 1.0;
134     else if ((ch>='j' && ch<='k') || (ch>='J' && ch<='K'))
135         group = 1.1;
136     else if ((ch>='l' && ch<='m') || (ch>='L' && ch<='M'))
137         group = 1.2;
138     else if ((ch>='n' && ch<='o') || (ch>='N' && ch<='O'))
139         group = 2.0;
140     else if ((ch>='p' && ch<='r') || (ch>='P' && ch<='R'))
141         group = 2.1;
142     else if ((ch>='s' && ch<='t') || (ch>='S' && ch<='T'))
143         group = 2.2;
144     else if ((ch>='u' && ch<='w') || (ch>='U' && ch<='W'))
145         group = 3.0;
146     else if ((ch>='x' && ch<='y') || (ch>='X' && ch<='Y'))
147         group = 3.1;
148     else if ((ch=='z') || (ch=='Z'))
149         group = 3.2;
150 
151     return group;
152 }
153 
154 /* Groups characters by using switch clauses */
155 double switchClauses(char ch)
156 {
157     double group;
158 
159     switch (ch) {
160         case 'a':
161         case 'b':
162         case 'c':
163         case 'A':
164         case 'B':
165         case 'C':
166             group = 0.0;
167             break;
168         case 'd':
```

```
169             case 'e':
170             case 'D':
171             case 'E':
172                 group = 0.1;
173                 break;
174             case 'f':
175             case 'g':
176             case 'F':
177             case 'G':
178                 group = 0.2;
179                 break;
180             case 'h':
181             case 'i':
182             case 'H':
183             case 'I':
184                 group = 1.0;
185                 break;
186             case 'j':
187             case 'k':
188             case 'J':
189             case 'K':
190                 group = 1.1;
191                 break;
192             case 'l':
193             case 'm':
194             case 'L':
195             case 'M':
196                 group = 1.2;
197                 break;
198             case 'n':
199             case 'o':
200             case 'N':
201             case 'O':
202                 group = 2.0;
203                 break;
204             case 'p':
205             case 'q':
206             case 'r':
207             case 'P':
208             case 'Q':
209             case 'R':
210                 group = 2.1;
211                 break;
212             case 's':
213             case 't':
214             case 'S':
215             case 'T':
216                 group = 2.2;
217                 break;
218             case 'u':
219             case 'v':
220             case 'w':
221             case 'U':
222             case 'V':
223             case 'W':
224                 group = 3.0;
```

```
225             break;
226         case 'x':
227         case 'y':
228         case 'X':
229         case 'Y':
230             group = 3.1;
231             break;
232         case 'z':
233         case 'Z':
234             group = 3.2;
235             break;
236         default:
237             group = NOT_IN_ALPHABET;
238     }
239
240     return group;
241 }
242
243 /* Groups characters by using user defined functions */
244 double userDefinedFunctions(char ch)
245 {
246     double group = NOT_IN_ALPHABET;
247
248     if (isGroup0(ch))
249         group = 0.0;
250     else if (isGroup1(ch))
251         group = 1.0;
252     else if (isGroup2(ch))
253         group = 2.0;
254     else if (isGroup3(ch))
255         group = 3.0;
256
257     return group;
258 }
259
260 /*  Checks if a given char is group 0 or not */
261 int isGroup0(char ch)
262 {
263     if ((ch>='a' && ch<='g') || (ch>='A' && ch<='G'))
264         return 1;
265
266     return 0;
267 }
268
269 /*  Checks if a given char is group 1 or not */
270 int isGroup1(char ch)
271 {
272     if ((ch>='h' && ch<='m') || (ch>='H' && ch<='M'))
273         return 1;
274
275     return 0;
276 }
277
278 /*  Checks if a given char is group 2 or not */
279 int isGroup2(char ch)
280 {
```

```
281     if ((ch>='n' && ch<='t') || (ch>='N' && ch<='T'))
282         return 1;
283 
284     return 0;
285 }
286 
287 /*  Checks if a given char is group 3 or not */
288 int isGroup3(char ch)
289 {
290     if ((ch>='u' && ch<='z') || (ch>='U' && ch<='Z'))
291         return 1;
292 
293     return 0;
294 }
```