```
/* Romulan word generator.  */
/* Created by Diane Duane,   */
/* ported to C by Curt Snyder  */
/**************************************/
#include stdlib
#include stdio
#include time

/* initialize ’r’ values */
char r[10][3] = {"H’","Ae","D’","W","U","N’","R’","O","V","Ll"};

/* initialize vowels */
char v[140][5] = {
"ae","A","i’","a"," Eh","e","T’","I","u’","s",
" ","ae","ea","ei","e","a","iu","o","ie","i",
"eo","i ","ae","ie","ai"," ","au","a","ei","ah",
"ao","a ","aeu","u"," ","ae","oa","i"," ","s",
"i",", ","ea","ia"," E","ei","ta’","ra-","ei"," ",
" ","’h","ae","oi","iy","u","ei","eh","s’h","i",
"e’","ia"," ","ie","iy","ih","ae","io","ai","ui",
"ae","y"," ","ei","ie","a’e","u","iu","ou","aa",
"a","i","ih","i’h","e ","ea","aa","ae","u","aeih",
"ae","ei"," -- ","iu","oa","ei","o","oi","ue"," ",
"’ss","l’","k’h","hw","rr","r","rr’","mm","t’d","’hh",
"qh","vh","fv","nh","d","e","hh","k","a","t",
"dl","dl","rh","nnh","rai","th","dh-","yrh","aith","qh",
"m","t","r","q","s","f","v","h","z","y"};

/* initialize consonants */
char c[140][5]={
"s","ll","R","m","k","t","h","r","rr","...",
" ","v","mn","kh","d","hv","fv"," ","r","t",
"-","th","k","lh","d","bh"," ","d’","dr","ht",
" ","ll","lh","dt’","ht","th","kh","l’","nn","n",
" ","’rh","rh","jh","kj","lh","nv","tr","hw","fv",
"nn","hw","d","nv","! ","mn","dh","rh","ll’","sw",
"lmn","l","mn","-","h’n","t","ss","hv","hs","hr",
"hj"," ","hf","wh","rrh","bh","j","y"," ","; ",
"llu","dh","kh","rh"," ",", ","; ","wh"," ","mn",
"e","ii","a","ee","eu","i","o","iu","uu"," ",
"uy","ae","e"," i","’i","’u","’u","iae","eu","a",
"ae","hl","iu","-a","ss","-t","r-","nn"," ’nh","ai",
"iu","iu","hu","ha","la","se","mo","tha","kha","dha",
"a"," i","t","e","e","ae","ai","ia","ia","ou"};

/* setup for randomization procedures */
int seed;
int rand6,rand2;

/* print an ’r’ field */
void print_rfield()
{
 int ivalue;

 ivalue = rand() % 10;
 printf ("%s",r[ivalue] );

}

/* print a vowel. Offset is the current index number */
void print_vowel( offset )
int offset;
{
 int ivalue;

 ivalue = rand() % 10 + (offset * 10);
 printf ("%s",v[ivalue] );
 if ( offset == 5 )
  rand6 = ivalue; /* save the random value, use it later */

}

/* additional vowel adjustment */
void more_adjustment()
{
 int ivalue;

 ivalue = rand () % 10 + 130;
 printf ("%s",v[ivalue]);
}


/* print out a consonant. Offset is the current index */
void print_consonant( offset )
int offset;
{
 int ivalue;

 ivalue = rand() % 10 + (offset * 10 );
 printf ("%s",c[ivalue] );
 if ( offset == 1 )
  rand2 = ivalue; /* save the random value, use it later */

}

/* end of word (maybe) */
void word_break()
{
 int ivalue;

 ivalue = rand() % 10;
 if (ivalue < 4) printf (" ");
}

/* adjustment when finished */
void terminal_adjustment()
{
 switch ( rand2 ) /* use the saved random value */
 {
  case 0 :
```

```
  case 1 :
  case 2 :
  case 3 :
  case 4 : printf (" ");
          break;
  case 5 : printf ("?");
          break;
  case 6 : printf ("....");
          break;
  case 7 : printf (", ");
          break;
  case 8 : printf (" ");
          break;
 }
}

/* generate text */
void gen_text()
{
 int t,g,i = 0;

 for ( t = 1; t<20; t++)
 {
  if ( g != 1 ) /* do this on the first pass */
  {
   g = 1;
   print_rfield();
  } /* g != 1 */
  for ( i = 0; i < 11; i++ ) /* print out 11 vowel/consonant/word breaks*/
  {
   print_vowel ( i );
   if ( i == 2 )
    word_break();
   print_consonant ( i );
   if ( ( i == 7 ) && ( rand6 >= 6 ))
    printf (" ");
  }
  more_adjustment();
  terminal_adjustment();
  printf ("\n"); /* line feed */
 } /* t loop */
 printf ("\n"); /* line feed */
} /* procedure gen_text */

/* initialize the random number generator, use session time as a
pseudo-random seed. If your compiler has a better random number
generator, USE IT! */
void initialize_random()
{
 struct tbuffer buffer;

 times ( &buffer );
 seed = buffer.proc_user_time;
 srand ( seed );
}
```

```
main()
{
 initialize_random();
 gen_text();  /* generate text */
}
```