

```
1. /* eliza.c
2. *
3. * original code by Weizenbaum, 1966
4. * this rendition based on Charles Hayden's Java implementation from http://chayden.net/eliza/Eliza.html
5. *
6. * Note: There are certainly far more optimal and elegant ways to code this... we kept this
7. * structure to be faithful to the original. -scaz
8. */
9.
10. #include <ctype.h>
11. #include <stdio.h>
12. #include <string.h>
13. #include <cs50.h>
14.
15. #define NUMKEYWORDS 37
16. #define MAXLINELEN 80
17. #define NUMSWAPS 14
18.
19. char *keywords[]= {
20.     "CAN YOU", "CAN I", "YOU ARE", "YOU'RE", "I DONT", "I FEEL",
21.     "WHY DONT YOU", "WHY CANT I", "ARE YOU", "I CANT", "I AM", "IM ",
22.     "YOU ", "I WANT", "WHAT", "HOW", "WHO", "WHERE",
23.     "WHEN", "WHY",
24.     "NAME", "CAUSE", "SORRY", "DREAM", "HELLO", "HI ", "MAYBE",
25.     " NO", "YOUR", "ALWAYS", "THINK", "ALIKE", "YES", "FRIEND",
26.     "COMPUTER", "CAR", "NOKEYFOUND"};
27.
28. char *SWAPS[NUMSWAPS][2] = {
29.     {"ARE", "AM"},
30.     {"WERE", "WAS"},
31.     {"YOU", "I"},
32.     {"YOUR", "MY"},
33.     {"IVE", "YOU'VE"},
34.     {"IM", "YOU'RE"},
35.     {"YOU", "ME"},
36.     {"ME", "YOU"},
37.     {"AM", "ARE"},
38.     {"WAS", "WERE"},
39.     {"I", "YOU"},
40.     {"MY", "YOUR"},
41.     {"YOUVE", "I'VE"},
42.     {"YOU'RE", "I'M"}
43. };
44.
45. int ResponsesPerKeyword[NUMKEYWORDS]= {
46.     3,2,4,4,4,3,
47.     3,2,3,3,4,4,
48.     3,5,9,9,9,9,
```

```
49.     9,9,
50.     2,4,4,4,1,1,5,
51.     5,2,4,3,7,3,6,
52.     7,5,6};
53.
54. char *responses[NUMKEYWORDS][9] = {
55.     { "DON'T YOU BELIEVE THAT I CAN*",
56.       "PERHAPS YOU WOULD LIKE TO BE ABLE TO*",
57.       "YOU WANT ME TO BE ABLE TO*"},
58.     { "PERHAPS YOU DON'T WANT TO*",
59.       "DO YOU WANT TO BE ABLE TO*"},
60.     { "WHAT MAKES YOU THINK I AM*",
61.       "DOES IT PLEASE YOU TO BELIEVE I AM*",
62.       "PERHAPS YOU WOULD LIKE TO BE*",
63.       "DO YOU SOMETIMES WISH YOU WERE*"},
64.     { "WHAT MAKES YOU THINK I AM*",
65.       "DOES IT PLEASE YOU TO BELIEVE I AM*",
66.       "PERHAPS YOU WOULD LIKE TO BE*",
67.       "DO YOU SOMETIMES WISH YOU WERE*"},
68.     { "DON'T YOU REALLY*",
69.       "WHY DON'T YOU*",
70.       "DO YOU WISH TO BE ABLE TO*",
71.       "DOES THAT TROUBLE YOU?"},
72.     { "TELL ME MORE ABOUT SUCH FEELINGS.",
73.       "DO YOU OFTEN FEEL*",
74.       "DO YOU ENJOY FEELING*"},
75.     { "DO YOU REALLY BELIEVE I DON'T*",
76.       "PERHAPS IN GOOD TIME I WILL*",
77.       "DO YOU WANT ME TO*"},
78.     { "DO YOU THINK YOU SHOULD BE ABLE TO*",
79.       "WHY CAN'T YOU*"},
80.     { "WHY ARE YOU INTERESTED IN WHETHER OR NOT I AM*",
81.       "WOULD YOU PREFER IF I WERE NOT*",
82.       "PERHAPS IN YOUR FANTASIES I AM*"},
83.     { "HOW DO YOU KNOW YOU CAN'T*",
84.       "HAVE YOU TRIED?",
85.       "PERHAPS YOU CAN NOW*"},
86.     { "DID YOU COME TO ME BECAUSE YOU ARE*",
87.       "HOW LONG HAVE YOU BEEN*",
88.       "DO YOU BELIEVE IT IS NORMAL TO BE*",
89.       "DO YOU ENJOY BEING*"},
90.     { "DID YOU COME TO ME BECAUSE YOU ARE*",
91.       "HOW LONG HAVE YOU BEEN*",
92.       "DO YOU BELIEVE IT IS NORMAL TO BE*",
93.       "DO YOU ENJOY BEING*"},
94.     { "WE WERE DISCUSSING YOU-- NOT ME.",
95.       "OH, I*",
96.       "YOU'RE NOT REALLY TALKING ABOUT ME, ARE YOU?"},
```

```
97.     { "WHAT WOULD IT MEAN TO YOU IF YOU GOT*",
98.       "WHY DO YOU WANT*",
99.       "SUPPOSE YOU SOON GOT*",
100.      "WHAT IF YOU NEVER GOT*",
101.      "I SOMETIMES ALSO WANT*"}},
102.     { "WHY DO YOU ASK?",
103.       "DOES THAT QUESTION INTEREST YOU?",
104.       "WHAT ANSWER WOULD PLEASE YOU THE MOST?",
105.       "WHAT DO YOU THINK?",
106.       "ARE SUCH QUESTIONS ON YOUR MIND OFTEN?",
107.       "WHAT IS IT THAT YOU REALLY WANT TO KNOW?",
108.       "HAVE YOU ASKED ANYONE ELSE?",
109.       "HAVE YOU ASKED SUCH QUESTIONS BEFORE?",
110.       "WHAT ELSE COMES TO MIND WHEN YOU ASK THAT?"},
111.     { "WHY DO YOU ASK?",
112.       "DOES THAT QUESTION INTEREST YOU?",
113.       "WHAT ANSWER WOULD PLEASE YOU THE MOST?",
114.       "WHAT DO YOU THINK?",
115.       "ARE SUCH QUESTIONS ON YOUR MIND OFTEN?",
116.       "WHAT IS IT THAT YOU REALLY WANT TO KNOW?",
117.       "HAVE YOU ASKED ANYONE ELSE?",
118.       "HAVE YOU ASKED SUCH QUESTIONS BEFORE?",
119.       "WHAT ELSE COMES TO MIND WHEN YOU ASK THAT?"},
120.     { "WHY DO YOU ASK?",
121.       "DOES THAT QUESTION INTEREST YOU?",
122.       "WHAT ANSWER WOULD PLEASE YOU THE MOST?",
123.       "WHAT DO YOU THINK?",
124.       "ARE SUCH QUESTIONS ON YOUR MIND OFTEN?",
125.       "WHAT IS IT THAT YOU REALLY WANT TO KNOW?",
126.       "HAVE YOU ASKED ANYONE ELSE?",
127.       "HAVE YOU ASKED SUCH QUESTIONS BEFORE?",
128.       "WHAT ELSE COMES TO MIND WHEN YOU ASK THAT?"},
129.     { "WHY DO YOU ASK?",
130.       "DOES THAT QUESTION INTEREST YOU?",
131.       "WHAT ANSWER WOULD PLEASE YOU THE MOST?",
132.       "WHAT DO YOU THINK?",
133.       "ARE SUCH QUESTIONS ON YOUR MIND OFTEN?",
134.       "WHAT IS IT THAT YOU REALLY WANT TO KNOW?",
135.       "HAVE YOU ASKED ANYONE ELSE?",
136.       "HAVE YOU ASKED SUCH QUESTIONS BEFORE?",
137.       "WHAT ELSE COMES TO MIND WHEN YOU ASK THAT?"},
138.     { "WHY DO YOU ASK?",
139.       "DOES THAT QUESTION INTEREST YOU?",
140.       "WHAT ANSWER WOULD PLEASE YOU THE MOST?",
141.       "WHAT DO YOU THINK?",
142.       "ARE SUCH QUESTIONS ON YOUR MIND OFTEN?",
143.       "WHAT IS IT THAT YOU REALLY WANT TO KNOW?",
144.       "HAVE YOU ASKED ANYONE ELSE?",
```

```
145.     "HAVE YOU ASKED SUCH QUESTIONS BEFORE?",
146.     "WHAT ELSE COMES TO MIND WHEN YOU ASK THAT?"},
147.     {
148.     "WHY DO YOU ASK?",
149.     "DOES THAT QUESTION INTEREST YOU?",
150.     "WHAT ANSWER WOULD PLEASE YOU THE MOST?",
151.     "WHAT DO YOU THINK?",
152.     "ARE SUCH QUESTIONS ON YOUR MIND OFTEN?",
153.     "WHAT IS IT THAT YOU REALLY WANT TO KNOW?",
154.     "HAVE YOU ASKED ANYONE ELSE?",
155.     "HAVE YOU ASKED SUCH QUESTIONS BEFORE?",
156.     "WHAT ELSE COMES TO MIND WHEN YOU ASK THAT?"},
157.     {
158.     "NAMES DON'T INTEREST ME.",
159.     "I DON'T CARE ABOUT NAMES-- PLEASE GO ON."},
160.     {
161.     "IS THAT THE REAL REASON?",
162.     "DON'T ANY OTHER REASONS COME TO MIND?",
163.     "DOES THAT REASON EXPLAIN ANY THING ELSE?",
164.     "WHAT OTHER REASONS MIGHT THERE BE?"},
165.     {
166.     "PLEASE DON'T APOLOGIZE.",
167.     "APOLOGIES ARE NOT NECESSARY.",
168.     "WHAT FEELINGS DO YOU HAVE WHEN YOU APOLOGIZE?",
169.     "DON'T BE SO DEFENSIVE!"},
170.     {
171.     "WHAT DOES THAT DREAM SUGGEST TO YOU?",
172.     "DO YOU DREAM OFTEN?",
173.     "WHAT PERSONS APPEAR IN YOUR DREAMS?",
174.     "ARE YOU DISTURBED BY YOUR DREAMS?"},
175.     {
176.     "HOW DO YOU DO--PLEASE STATE YOUR PROBLEM."},
177.     {
178.     "HOW DO YOU DO--PLEASE STATE YOUR PROBLEM."},
179.     {
180.     "YOU DON'T SEEM QUITE CERTAIN.",
181.     "WHY THE UNCERTAIN TONE?",
182.     "CAN'T YOU BE MORE POSITIVE?",
183.     "YOU AREN'T SURE?",
184.     "DON'T YOU KNOW?"},
185.     {
186.     "ARE YOU SAYING NO JUST TO BE NEGATIVE?",
187.     "YOU ARE BEING A BIT NEGATIVE.",
188.     "WHY NOT?",
189.     "ARE YOU SURE?",
190.     "WHY NO?"},
191.     {
192.     "WHY ARE YOU CONCERNED ABOUT MY*",
193.     "WHAT ABOUT YOUR OWN*"},
194.     {
195.     "CAN YOU THINK OF A SPECIFIC EXAMPLE?",
196.     "WHEN?",
197.     "WHAT ARE YOU THINKING OF?",
198.     "REALLY, ALWAYS?"},
199.     {
200.     "DO YOU REALLY THINK SO?",
201.     "BUT YOU ARE NOT SURE YOU*",
202.     "DO YOU DOUBT YOU*"},
203.     {
204.     "IN WHAT WAY?",
205.     "WHAT RESEMBLANCE DO YOU SEE?",
```

```
193.         "WHAT DOES THE SIMILARITY SUGGEST TO YOU?",
194.         "WHAT OTHER CONNECTIONS DO YOU SEE?",
195.         "COULD THERE REALLY BE SOME CONNECTION?",
196.         "HOW?"},
197.     { "YOU SEEM QUITE POSITIVE.",
198.       "ARE YOU SURE?",
199.       "I SEE.",
200.       "I UNDERSTAND."},
201.     { "WHY DO YOU BRING UP THE TOPIC OF FRIENDS?",
202.       "DO YOUR FRIENDS WORRY YOU?",
203.       "DO YOUR FRIENDS PICK ON YOU?",
204.       "ARE YOU SURE YOU HAVE ANY FRIENDS?",
205.       "DO YOU IMPOSE ON YOUR FRIENDS?",
206.       "PERHAPS YOUR LOVE FOR FRIENDS WORRIES YOU?"},
207.     { "DO COMPUTERS WORRY YOU?",
208.       "ARE YOU TALKING ABOUT ME IN PARTICULAR?",
209.       "ARE YOU FRIGHTENED BY MACHINES?",
210.       "WHY DO YOU MENTION COMPUTERS?",
211.       "WHAT DO YOU THINK MACHINES HAVE TO DO WITH YOUR PROBLEM?",
212.       "DON'T YOU THINK COMPUTERS CAN HELP PEOPLE?",
213.       "WHAT IS IT ABOUT MACHINES THAT WORRIES YOU?"},
214.     { "OH, DO YOU LIKE CARS?",
215.       "MY FAVORITE CAR IS A LAMBORGINI COUNTACH. WHAT IS YOUR FAVORITE      CAR?",
216.       "MY FAVORITE CAR COMPANY IS FERRARI. WHAT IS YOURS?",
217.       "DO YOU LIKE PORSCHEs?",
218.       "DO YOU LIKE PORSCHE TURBO CARRERAS?"},
219.     { "SAY, DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?",
220.       "WHAT DOES THAT SUGGEST TO YOU?",
221.       "I SEE.",
222.       "I'M NOT SURE I UNDERSTAND YOU FULLY.",
223.       "COME, COME ELUCIDATE YOUR THOUGHTS.",
224.       "CAN YOU ELABORATE ON THAT?",
225.       "THAT IS QUITE INTERESTING."}
226.
227.
228. };
229.
230. void print_center(char *msg) {
231.     int numspaces=(MAXLINELEN-strlen(msg))/2;
232.
233.     for(int i=0;i<numspaces;i++)
234.         printf(" ");
235.     printf("%s\n", msg);
236.     return;
237. }
238.
239. void print_title () {
240.     printf("\n\n");
```

```
241.     print_center("*** ELIZA ***");
242.     print_center("Original code by Weizenbaum, 1966");
243.     print_center("To stop Eliza, type 'bye'");
244.     printf("\n\n");
245.     printf("HI!  I'M ELIZA.  WHAT'S YOUR PROBLEM?\n");
246. }
247.
248. void readline(char *instr) {
249.     char c;
250.     int slen=0;
251.
252.     c=getchar();
253.     while (c != '\n')
254.     {
255.         // removes punctuation and sets to uppercase
256.         if(isalpha(c) || isspace(c))
257.             instr[slen++]=toupper(c);
258.         if(slen>MAXLINELEN-1)
259.         {
260.             printf("Exceeded Max Line Length\n");
261.             exit(0);
262.         }
263.         c=getchar();
264.     }
265.     instr[slen]='\0';
266. }
267.
268.
269. int main(int argc, char * argv[])
270. {
271.     int k,baseLength;
272.     int whichReply[NUMKEYWORDS];
273.     char lastinput[MAXLINELEN];
274.     char reply[MAXLINELEN];
275.     char *baseResponse, *token;
276.     const char separator[2]=" ";
277.     char inputstr[MAXLINELEN];
278.
279.     // use the first reply for each keyword match the first time you see that keyword
280.     for (int x=0;x< NUMKEYWORDS; x++) {
281.         whichReply[x] = 0;
282.     }
283.
284.     // print a nice centered title screen
285.     print_title();
286.
287.     lastinput[0]='\0';
288.
```

```
289.     while (1) {
290.         readline(inputstr);
291.
292.         // check for termination
293.         if (strcmp(inputstr,"BYE")==0)
294.             break;
295.
296.         // check for repeated entries
297.         if (strcmp(lastinput,inputstr)==0)
298.             {
299.                 printf( "PLEASE DON'T REPEAT YOURSELF!\n");
300.                 continue;
301.             }
302.         strncpy(lastinput,inputstr,strlen(inputstr)+1);
303.
304.         // see if any of the keywords is contained in the input
305.         // if not, we use the last element of keywords as our default responses
306.         char *location;
307.         strcpy(reply,"");
308.         for(k=0;k<NUMKEYWORDS-1;k++)
309.             {
310.                 location=strstr(inputstr, keywords[k]);
311.                 if(location != NULL)
312.                     break;
313.             }
314.
315.         // Build Eliza's response
316.         // start with Eliza's canned response, based on the keyword match
317.         baseResponse = responses[k][whichReply[k]];
318.         baseLength = strlen(baseResponse);
319.
320.         if(baseResponse[baseLength-1] != '*')
321.             {
322.                 // if we have a baseResponse without an asterix, just use it as-is
323.                 strcat(reply, baseResponse);
324.             }
325.         else
326.             {
327.                 // if we do have an asterix, fill in the remaining with the user input
328.                 // use all but the last character of the base response
329.                 strcat(reply, baseResponse, baseLength-1);
330.
331.                 // now add in the rest of the user's input, starting at <location>
332.                 // but skip over the keyword itself
333.                 location+=strlen(keywords[k]);
334.                 // take them one word at a time, so that we can substitute pronouns
335.                 token = strtok(location, separator);
336.                 while(token != NULL)
```

```
337.     {
338.         for(int s=0;s<NUMSWAPS;s++)
339.         {
340.             if(strcmp(SWAPS[s][0], token) == 0)
341.             {
342.                 token=SWAPS[s][1];
343.                 break;
344.             }
345.         }
346.         strcat(reply, " ");
347.         strcat(reply, token);
348.         token=strtok(NULL, separator);
349.     };
350.     strcat(reply, "?");
351. }
352. printf("%s\n", reply);
353.
354. // next time, use the next appropriate reply for that keyword
355. whichReply[k]++;
356. if ( whichReply[k] >= ResponsesPerKeyword[k])
357.     whichReply[k] = 0;
358.
359. }
360.
361. printf( "GOODBYE!  THANKS FOR VISITING WITH ME...\n");
362.
363. }
```