

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <ctype.h>
4  extern double atof();
5  #include "cmdline.h"
6
7  static struct entrada_opcion tabla_opciones[] = {
8      {"-?",          2, AYUDA},
9      {"-ayuda",      2, AYUDA},
10     {"-medida",      3, MEDIDA},
11     {"GKOM",         3, GKOM},
12     {"LKOM",         3, LKOM},
13     {"LKHM",         3, LKHM},
14     {"GKHM",         3, GKHM},
15     {"LIHE",         2, LIHE},
16     {"GIHE",         2, GIHE},
17     {"-todas",       2, TODAS},
18     {"-maximo",      3, MAX},
19     {"-promedio",    3, PROMEDIO},
20     {"-minimo",      3, MIN},
21     {"-inferior",    3, INF},
22     {"-superior",    3, SUP},
23 };
24 static int tamano_tabla_opciones = sizeof(tabla_opciones) /
25                                     sizeof(struct entrada_opcion) - 1;
26
27 void uso(progn)
28     char *progn;
29 {
30     fprintf(stderr,
31         "Uso: %s -medida <MEDIDA> [ Opcion-busquea ] Fichero [ Fichero ... ]
32     \n", progn);
33     exit(1);
34 }
35 static int cadena_es_opcion(arg, arg_valido, long_min_req)
36     char *arg, *arg_valido;
37     int long_min_req;
38 {
39     int long_arg, res;
40
41     res = 0;
42     long_arg = strlen(arg);
43     if ((strncmp(arg, arg_valido, long_arg) == 0) &&
44         (long_arg >= long_min_req))
45         res = 1;
46
47     return res;
48 }
49 static int codigo_cadena(cadena)
50     char *cadena;
51 {
52     int i, res;
53
54     res = -1;
55     for (i = 1; i < tamano_tabla_opciones; ++i) {
56         if (cadena_es_opcion(cadena, tabla_opciones[i].opcion,
57             tabla_opciones[i].long_min)) {
58             res = tabla_opciones[i].identificador;
59             break;

```

```

59     }
60 }
61 return res;
62 }
63
64 void imprimir_comando(comando)
65     struct comando *comando;
66 {
67
68     if ((comando->medida != NULL && strcmp(comando->medida, "-ayuda")==0)
69         || (comando->medida == NULL && comando->busqueda == NULL)) {
70         uso("cmdline");
71         return;
72     }
73
74     printf("Resumen del comando:\n");
75     if (comando->medida)
76         printf("Medida:  %s\n", comando->medida);
77     if (comando->busqueda) {
78         printf("Busqueda: %s\n", comando->busqueda);
79         if (strcmp(comando->busqueda, "-inferior") == 0 ||
80             strcmp(comando->busqueda, "-superior") == 0)
81             printf("Numero:  %.1f\n", comando->numero);
82     }
83
84 void imprimir_ficherosin(argi, argc, argv)
85     int argi, argc;
86     char **argv;
87 {
88     int i;
89
90     if (argc == argi)
91         printf("No se han proporcionado ficheros de entrada\n");
92     else
93     {
94         printf("Numero de ficheros de entrada: %d\n", argc - argi);
95         printf("Los ficheros de entrada son:\n");
96         for (i ; i < argc; ++i)
97             printf("    %s\n", argv[i]);
98     }
99 }
100
101 int procesar_opciones(argc, argv, comando)
102     int argc;
103     char **argv;
104     struct comando *comando;
105 {
106     int i, res = 0, posicion = 0;
107
108     comando->numero = 0.0;
109     comando->medida = NULL;
110     comando->busqueda = NULL;
111
112     for (i = 1; i < argc; ++i) {
113         switch (codigo_cadena(argv[i])) {
114
115             case AYUDA:
116                 comando->medida = "-ayuda";
117                 break;

```

```

118
119     case MEDIDA:
120         if (comando->medida != NULL) {
121             fprintf(stderr, "Demasiadas medidas\n");
122             res = -1;
123         }
124         else {
125             if (++i >= argc) {
126                 fprintf(stderr, "No hay medidas\n");
127                 res = -1;
128             }
129             else {
130                 switch (codigo_cadena(argv[i])) {
131                     case GKOM:
132                         comando->medida = "GKOM";
133                         break;
134                     case LKOM:
135                         comando->medida = "LKOM";
136                         break;
137                     case LKHM:
138                         comando->medida = "LKOM";
139                         break;
140                     case GKHM:
141                         comando->medida = "GKHM";
142                         break;
143                     case LIHE:
144                         comando->medida = "LIHE";
145                         break;
146                     case GIHE:
147                         comando->medida = "GIHE";
148                         break;
149                     default:
150                         fprintf(stderr, "Opcion de medida no valida\n");
151                         res = -1;
152                         break;
153                 }
154             }
155         }
156         break;
157
158     case TODAS:
159         if (comando->busqueda != NULL) {
160             fprintf(stderr, "Demasiadas opciones de busqueda\n");
161             res = -1;
162         }
163         else
164             comando->busqueda = "-todas";
165         break;
166
167     case MAX:
168         if (comando->busqueda != NULL) {
169             fprintf(stderr, "Demasiadas opciones de busqueda\n");
170             res = -1;
171         }
172         else
173             comando->busqueda = "-maximo";
174         break;
175
176     case PROMEDIO:
177         if (comando->busqueda != NULL) {
178             fprintf(stderr, "Demasiadas opciones de busqueda\n");

```

```

179         res = -1;
180     }
181     else
182         comando->busqueda = "-promedio";
183     break;
184
185 case MIN:
186     if (comando->busqueda != NULL) {
187         fprintf(stderr, "Demasiadas opciones de busqueda\n");
188         res = -1;
189     }
190     else
191         comando->busqueda = "-minimo";
192
193 case INF:
194     if (comando->busqueda != NULL) {
195         fprintf(stderr, "Demasiadas opciones de busqueda\n");
196         res = -1;
197     }
198     else {
199         comando->busqueda = "-inferior";
200         ++i;
201         if (i >= argc || !isdigit(*argv[i])) {
202             fprintf(stderr, "No se ha proporcionado un numero\n");
203             res = -1;
204         }
205         else {
206             comando->numero = atof(argv[i]);
207         }
208     }
209     break;
210
211 case SUP:
212     if (comando->busqueda != NULL) {
213
214         res = -1;
215     }
216     else {
217         comando->busqueda = "-superior";
218         ++i;
219         if (i >= argc || !isdigit(*argv[i])) {
220             fprintf(stderr, "No se ha proporcionado un numero\n");
221             res = -1;
222         }
223         else {
224             comando->numero = atof(argv[i]);
225         }
226     }
227     break;
228
229 default:
230     posicion = i;
231     break;
232 }
233
234 if (res < 0)
235     break;
236 }
237
238 if (res < 0)
239     fprintf(stderr, "Los argumentos de la opcion son incorrectos\n");

```

```

240         else if (posicion == 0)
241             fprintf(stderr, "No se han proporcionado ficheros de
entrada\n");
242         else {
243             imprimir_comando(comando);
244             imprimir_ficherosin(posicion, argc, argv);
245         }
246         return res;
247     }
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266     /*****
267     *****/
268     Este es el programa principal.
269     No hay que crear datos de prueba para este, crear abstracciones, etc.
270     *****/
271     *****/
272     int main(argc, argv)
273         int argc;
274         char **argv;
275     {
276         struct comando a;
277         int res;
278
279         if (argc == 1) {
280             uso("cmdline");
281             res = -1;
282         }
283         else
284             res = procesar_opciones(argc, argv, &a);
285
286         return res;
287     }

```