

```
1  # include <stdio.h>
2  # include <stdlib.h>
3  # include <string.h>
4
5  #define MAX 50
6
7
8  typedef struct Data {
9      char num[MAX];
10     char name[MAX];
11     char s_name[MAX];
12     unsigned short int age;
13     char job[MAX];
14     float pay;
15 } DB;
16
17
18 int upload(char [], DB []);
19 void printDB(DB [], int);
20 void ageSelect(DB [], char [], int);
21 void IDSelect(DB [], char [], char [], int);
22 void jobSelect(DB [], char [], char [], int);
23
24
25
26 /*****
27  /* A menu to choose the operation to perform on the dataset. */
28
29 int main (int argc, char *argv[]) {
30     if (argc != 3) {
31         printf("\nError: Invalid Argument Number!\n");
32         return 0;
33     }
34
35     printf("\nQueries:\n");
36     printf("\t1 - People older than n years of age.\n");
37     printf("\t2 - Personal data corresponding to an ID number.\n");
38     printf("\t3 - People with the same occupation.\n");
39
40     unsigned short int choice;
41     printf("\nQuery: ");
42     scanf("%d", &choice);
43
44     DB dbData[MAX];
45     int dbSize = upload( *(argv + 1) , dbData);
46     char w[MAX];
47
48     if (choice == 1) {
49         getchar();
50         ageSelect(dbData , *(argv + 2) , dbSize);
51     }
52     else if (choice == 2) {
53         printf ("ID number: ");
54         getchar();
55         gets(w);
56         IDSelect( dbData , *(argv + 2) , w , dbSize);
57     }
58 }
```

```

59
60     else if (choice == 3) {
61         printf("Occupation: ");
62         getchar();
63         gets(w);
64         jobSelect(dbData, *(argv + 2), w, dbSize);
65     }
66     else {
67         printf ("Incorrect Query!\n");
68         return 0;
69     }
70     printf ("\nResults in %s", *(argv + 2));
71 }
72
73
74 /*****
75
76 /* Reads a dataset from a text file. */
77
78 int upload (char s[], DB dbData []) {
79     FILE *fin;
80     int i = 0;
81
82     fin = fopen (s, "r");
83
84     while ( fscanf(fin, "%s %s %s %d %s %f\n", &dbData[i].num      , &dbData[i].name      ,
85                                                &dbData[i].s_name , &dbData[i].age      ,
86                                                &dbData[i].job      , &dbData[i].pay) == 6
87                                                && i < MAX)
88         i++;
89     return i;
90 }
91
92
93 /*****
94
95 /* Prints the dataset on a text file - if needed for some reason...*/
96
97 void printDB(DB sorted[], int i) {
98     FILE * fout = fopen("dbRead.txt", "w");
99
100     for (int j = 0; j < i; j++)
101         fprintf(fout, "%4s %-20s %-20s %3d %-15s %8.2f\n",
102                sorted[j].num, sorted[j].name,
103                sorted[j].s_name, sorted[j].age,
104                sorted[j].job, sorted[j].pay);
105 }
106
107
108
109
110
111
112
113
114
115
116

```

```

117 /*****
118
119  */ Extract all the entries basing on the age given as input at runtime. */
120
121 void ageSelect(DB dbData[], char fileOut[], int dbSize) {
122     unsigned short int age;
123     printf("\nAge: ");
124     scanf("%d" , &age);
125
126     DB sorted[MAX];
127     FILE * fout;
128     fout = fopen(fileOut , "w");
129     int i = 0;
130
131     for (int j = 0 ; j < dbSize ; j++) {
132         if (dbData[j].age >= age)
133             sorted[i++] = dbData[j];
134     }
135
136     if (i == 0) {
137         printf("No match for age >= %d\n", age);
138         fprintf(fout, "No match for age ≥ %d", age);
139     }
140     else if (i == 1) {
141         fprintf(fout , "A single match for age ≥ %d:\n\n", age);
142         fprintf(fout , "%4s %-20s %-20s %3d %-15s %8.2f\n",
143             sorted[0].num , sorted[0].name,
144             sorted[0].s_name , sorted[0].age,
145             sorted[0].job , sorted[0].pay );
146     }
147     else {
148         fprintf(fout, "%d matches for age ≥ %d:\n\n", i, age);
149         for (int j = 0 ; j < i ; j++)
150             fprintf(fout , "%4s %-20s %-20s %3d %-15s %8.2f\n",
151                 sorted[j].num , sorted[j].name,
152                 sorted[j].s_name , sorted[j].age,
153                 sorted[j].job , sorted[j].pay );
154     }
155 }
156 }
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174

```

```

175 /*****
176
177  /* Search for a single entry for a given ID. */
178
179  /* IMPORTANT: Assumes that the elements are sorted by ID number because it performs a
180      binary search in the dataset. */
181
182  void IDSelect(DB dbData [], char fileOut[], char ID[], int dbSize) {
183      FILE * fout;
184      fout = fopen(fileOut, "w");
185
186      int cond, low = 0, high, mid;
187      high = dbSize - 1;
188
189      while (low <= high) {
190          mid = (low + high) / 2;
191          if ( (cond = strcmp (ID, dbData[mid].num)) < 0 )
192              high = mid - 1;
193          else if (cond > 0)
194              low = mid + 1;
195          else { /*found*/
196              fprintf(fout, "Data for ID = %s:\n\n", ID);
197              fprintf(fout, "%-20s %-20s %3d    %-15s %8.2f",
198                  dbData[mid].name , dbData[mid].s_name , dbData[mid].age,
199                  dbData[mid].job   , dbData[mid].pay);
200              return;
201          }
202      }
203      printf("No match found.\n");
204      fprintf(fout, "\nNo match found.");
205  }
206
207
208
209
210 /*****
211
212  /* Select all entries with the same job. */
213
214
215  void jobSelect (DB dbData [], char fileOut[], char job[], int dbSize) {
216      FILE * fout;
217      fout = fopen (fileOut , "w");
218      DB positions[MAX];
219
220      /*Linear search*/
221      int k = 0;
222      for (int i = 0 ; i < dbSize ; i++) {
223          if ((strcmp (job, dbData[i].job)) == 0)
224              positions[k++] = dbData[i];
225      }
226
227      if (k == 0) {
228          printf("\nNo match found for \"%s\"\n", job);
229          fprintf(fout, "\nNo match found for \"%s\"\n", job);
230      }
231
232

```

```
233     else if (k == 1)
234         fprintf (fout, "\nOne match found for \"%s\"\n\n", job);
235     else
236         fprintf (fout, "%d matches found for \"%s\"\n\n", k , job);
237
238     for (int j = 0 ; j < k ; j++)
239         fprintf (fout, "%4s %-20s %-20s %3d %8.2f\n",
240                 positions[j].num , positions[j].name ,
241                 positions[j].s_name , positions[j].age ,
242                 positions[j].pay );
243 }
```