```
1 /*
2 * C program by Dave Russillo. Made for CS1311.
3 * Takes collection of items and puts it in linked list.
4 */
5#include <stdio.h>
6#include <stdlib.h>
7#include <string.h>
9 struct videogame {
   char name[31];
10
   double space; // in GB
11
   int playtime; // in hours
double price; // in USD
12
13
   struct videogame *next;
14
15 }:
16 struct videogame *head, *last, *current;
18 int entries; // length of list
19 double total space;
20 int total playtime;
21 double total price;
22
23
24
25 void add node(void) {
   int valid_value = 0; // boolean
26
27
   // new struct
28
   current = malloc(sizeof(struct videogame));
29
   // set name
31
```

```
printf("Enter name (will read up to 30 characters): ");
   fgets(current->name, 31, stdin);
33
   if(current->name[29] == '\n') { // if input is exactly 30
34
     current->name[29] == '\0';
35
   } else if(strlen(current->name) == 30) { // if input is greater than 30
36
     while(getchar() != '\n'); // clear input buffer
37
   } else {
38
     current->name[strlen(current->name)-1] = '\0'; // replace newline
39
   }
40
41
   // set space
42
   while(valid value == 0) {
43
     printf("Enter required space in GB (up to 9,999.9): ");
44
     scanf("%lf", &current->space);
45
     if(current->space < 0.000001 || current->space > 9999.9) {
46
       printf("Invalid value for required space was entered. Try again. \n");
47
48
     } else {
       valid value = 1; // pass
49
50
51
   total space += current->space;
52
   valid value = 0; // reset valid value
53
54
   // set playtime
55
   while(valid value == 0) {
56
     printf("Enter playtime in hours (up to 99,999): ");
57
     scanf("%d", &current->playtime);
58
     if(current->playtime < 0 || current->playtime > 99999) {
59
       printf("Invalid value for playtime was entered. Try again. \n");
60
     } else {
61
       valid value = 1; // pass
62
```

```
}
63
64
   total_playtime += current->playtime;
65
   valid value = 0; // reset valid value
66
67
68
   // set price
   while(valid value == 0) {
69
      printf("Enter price in USD (enter up to 999.99): ");
70
      scanf("%lf", &current->price);
71
     if(current->price < 0 || current->price > 999.99) {
72
        printf("Invalid value for price was entered. Try again. \n");
73
      } else {
74
        valid_value = 1; // pass
75
76
77
   total price += current->price;
78
   valid value = 0; // reset valid value
79
80
   while(getchar() != '\n'); // clear input buffer
81
82
83
   // set next
   current->next = NULL;
84
85
   if(head == NULL) {
86
     head = current;
87
   } else {
88
     last->next = current;
89
90
   last = current;
91
   entries++; // track entries
92
93 }
```

```
95
96
97 void delete node(void) {
    char target[31];
98
    int value_in_list = 0; // boolean default false
99
    struct videogame *prev; // tracks previous node
100
101
    while(value in list == 0) {
102
      printf("Enter name of target entry (type exit to go back): ");
103
      fgets(target, 31, stdin);
104
      if(strcspn(target, "\n") == 30) {
105
        while(getchar() != '\n');
106
      } else {
107
        target[strcspn(target, "\n")] = '\0';
108
109
      current = head;
110
      if(strcmp(target, "exit") == 0) {
        return;
      } else if(strcmp(current->name, target) == 0) {
        value in list = 1;
        head = head->next:
        free(current);
116
      } else {
117
        current = head->next;
118
        prev = head;
119
        while(current != NULL && strcmp(current->name, target) != 0) {
120
121
           prev = current;
122
           current = current->next;
123
        if(current == NULL) {
124
```

```
value_in_list = 0; // redundant for safety
125
           printf("Item not present. Try again. \n");
126
         127
           value in list = 1;
128
129
           prev->next = current->next;
130
           free(current);
131
132
133
134
135
136
137
138 void print_nodes(void) {
    char price_buffer[8]; // used for formatting price with $ sign in front
139
140
141
    printf("CURRENT DATA (%d ENTRIES): \n", entries);
142
    printf("
                                                                                           \n");
143
    printf("|
                                                                                           \n");
144
    printf("| name
                                                                                           \n");
145
                                                                   playtime
                                                                                price
                                                    req. space
                                                                                           \n");
146
    printf("|
    if(head == NULL) {
147
       printf("
                                                                                             \n");
148
       printf("| NULL
                                                             NULL
                                                                          NULL
                                                                                             \n");
149
                                                                                     NULL
       printf("|
150
                                                                                             \n");
151
    } else {
       current = head:
152
       while(current != NULL) {
153
         sprintf(price buffer, "$%.2f", current->price);
154
                                                                                             | \n");
155
       printf("| %-30s | %8.1fGB | %6dh | %7s | \n", current->name, current->space, current->playtime, price_buffer);
156
       printf("
157
158
       current = current->next;
159
     // print total values
160
     sprintf(price_buffer, "$%.2f", total_price);
161
162
     printf(
163
             Total entries: %12d
             req. space: %15.1f GB | \n"
164
             playtime: %17d h | \n"
165
                          | \n'"
166
             price: %20s
167
           entries, total space, total playtime, price buffer);
168
169
170
   printf("\n\n");
171 }
172
173
174
175 int main(void) {
   char selection = '0'; // add or delete
176
177
178
   printf("This program take videogames' names, required space, playtime, and price as input. \n"
179
         "It then puts them in a liked list database. \n\n"
180
         "Press enter to continue \n");
181
182
   getchar();
183
   while(selection != 'a' && selection != 'd' && selection != 'q') {
184
185
     print nodes();
     printf("What would you like to do? \n"
186
```

```
"Add Entry(a), Delete Entry(d), Quit(q)\n\n?..");
187
       scanf(" %c", &selection);
188
       getchar();
189
190
       if(selection == 'a') {
191
         print nodes();
192
         add node();
193
         selection = '0';
194
       } else if(selection == 'd') {
195
         print nodes();
196
         delete_node();
197
         selection = '0';
198
199
200
201
202
    return 0;
203
204
```

This program take videogames' names, required space, playtime, and price as input. It then puts them in a liked list database.

Press enter to continue

CURRENT DATA (0 ENTRIES):

 name 	req. space	playtime	price
 NULL 	NULL	NULL	NULL

What would you like to do?
Add Entry(a), Delete Entry(d), Quit(q)

?..a

CURRENT DATA (0 ENTRIES):

name			
	req. space	playtime	price
NULL	 NULL	 NULL 	 NULL

Enter name (will read up to 30 characters): Doom Eternal

Enter required space in GB (up to 9,999.9): 88.9

Enter playtime in hours (up to 99,999): 15

Enter price in USD (enter up to 999.99): 39.99

CURRENT DATA (1 ENTRIES):

name			
	req. space	playtime	price
Doom Eternal			
	88.9GB	15h	\$39.99

Total entries: 1
req. space: 88.9 GB
playtime: 15 h
price: \$39.99

What would you like to do?
Add Entry(a), Delete Entry(d), Quit(q)

?..a

CURRENT DATA (5 ENTRIES):

name			
	req. space	playtime	price
Doom Eternal			
	88.9GB	15h	\$39.99
Minecraft			
	0.6GB	200h	\$23.99
Grand Theft Auto V			
	110.4GB	1126h	\$10.99
Rainbow Six Siege			
	82.3GB	1312h	\$6.89
Garfield Kart			
	1.8GB	12h	\$14.99

Total entries: 5
req. space: 284.0 GB
playtime: 2665 h
price: \$96.85

What would you like to do?
Add Entry(a), Delete Entry(d), Quit(q)

?..d

CURRENT DATA (5 ENTRIES):

price:

 name 		 req. space _	 playtime 	price
 Doom Eternal 		88.9GB	15h	\$39.99
 Minecraft 		0.6GB	200h	\$23.99
 Grand Theft Auto V 		 110.4GB	1126h	\$10.99
 Rainbow Six Siege 		82.3GB	1312h	\$6.89
 Garfield Kart 		1.8GB	12h	\$14.99
 Total entries: req. space: playtime:	5 284.0 GB 2665 h			

Enter name of target entry (type exit to go back): Minecraft[

\$96.85

CURRENT DATA (5 ENTRIES):

 name 	 req. space 	 playtime 	 price
 Doom Eternal	 88.9GB 	15h	\$39.99
 Grand Theft Auto V 	 110.4GB	1126h	\$10.99
 Rainbow Six Siege 	 82.3GB	1312h	\$6.89
 Garfield Kart 	1.8GB	12h	\$14.99
Total entries: 5			

Total entries: 5
req. space: 284.0 GB
playtime: 2665 h
price: \$96.85

What would you like to do?
Add Entry(a), Delete Entry(d), Quit(q)

?..q