

```

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>
8
9  struct videogame {
10     char name[31];
11     double space; // in GB
12     int playtime; // in hours
13     double price; // in USD
14     struct videogame *next;
15 };
16 struct videogame *head, *last, *current;
17
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) {
26     int valid_value = 0; // boolean
27
28     // new struct
29     current = malloc(sizeof(struct videogame));
30
31     // set name

```

```

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

```

```

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

```

```

94
95
96
97 void delete_node(void) {
98     char target[31];
99     int value_in_list = 0; // boolean default false
100    struct videogame *prev; // tracks previous node
101
102    while(value_in_list == 0) {
103        printf("Enter name of target entry (type exit to go back): ");
104        fgets(target, 31, stdin);
105        if(strcspn(target, "\n") == 30) {
106            while(getchar() != '\n');
107        } else {
108            target[strcspn(target, "\n")] = '\0';
109        }
110        current = head;
111        if(strcmp(target, "exit") == 0) {
112            return;
113        } else if(strcmp(current->name, target) == 0) {
114            value_in_list = 1;
115            head = head->next;
116            free(current);
117        } else {
118            current = head->next;
119            prev = head;
120            while(current != NULL && strcmp(current->name, target) != 0) {
121                prev = current;
122                current = current->next;
123            }
124            if(current == NULL) {

```

[illegible]

```

187         "Add Entry(a), Delete Entry(d), Quit(q)\n\n?..");
188     scanf(" %c", &selection);
189     getchar();
190
191     if(selection == 'a') {
192         print_nodes();
193         add_node();
194         selection = '0';
195     } else if(selection == 'd') {
196         print_nodes();
197         delete_node();
198         selection = '0';
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):

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		

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

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			
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