```c
/*
 * C program by Dave Russillo. Made for CS1311.
 * Takes collection of items and puts it in linked list.
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct videogame {
  char name[31];
  double space;   // in GB
  int playtime;   // in hours
  double price;   // in USD
  struct videogame *next;
};
struct videogame *head, *last, *current;
int entries;   // length of list


void add_node(void) {
  int valid_value = 0;   // boolean

  // new struct
  current = malloc(sizeof(struct videogame));

  // set name
  printf("Enter name (will read up to 30 characters): ");
  fgets(current->name, 31, stdin);
  // remove newline
  if(strcspn(current->name, "\n") == 30) {  // if strcspn returns length, input exceeds limit
    while(getchar() != '\n');
```

```c
32    } else {
33      current->name[strcspn(current->name, "\n")] = '\0';
34    }
35
36    // set space
37    while(valid_value == 0) {
38      printf("Enter required space in GB (up to 9,999.9): ");
39      scanf("%lf", &current->space);
40      if(current->space < 0.000001 || current->space > 9999.9) {
41        printf("Invalid value for required space was entered. Try again. \n");
42      } else {
43        valid_value = 1;   // pass
44      }
45    }
46    valid_value = 0;   // reset valid_value
47
48    // set playtime
49    while(valid_value == 0) {
50      printf("Enter playtime in hours (up to 99,999): ");
51      scanf("%d", &current->playtime);
52      if(current->playtime < 0 || current->playtime > 99999) {
53        printf("Invalid value for playtime was entered. Try again. \n");
54      } else {
55        valid_value = 1;   // pass
56      }
57    }
58    valid_value = 0;   // reset valid_value
59
60    // set price
61    while(valid_value == 0) {
62      printf("Enter price in USD (enter up to 999.99): ");
```

```c
         scanf("%lf", &current->price);
         if(current->price < 0 || current->price > 999.99) {
            printf("Invalid value for price was entered. Try again. \n");
         } else {
            valid_value = 1;   // pass
         }
      }
      valid_value = 0;   // reset valid_value

      while(getchar() != '\n');   // clear input buffer

      // set next
      current->next = NULL;

      if(head == NULL) {
         head = current;
      } else {
         last->next = current;
      }
      last = current;
      entries++;   // track entries
}


void print_nodes(void) {
   char price_buffer[8];

   printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
   printf("CURRENT DATA (%d ENTRIES): \n", entries);
   printf("  _____  \n");
   printf("|                                      |          |        |         | \n");
```

```c
     printf("| name                                    | req. space | playtime |  price  | \n");
     printf("|_____|_____|_____|_____| \n");
     if(head == NULL) {
        printf("|                                         |            |          |         | \n");
        printf("|  NULL                                   |      NULL  |     NULL |    NULL | \n");
        printf("|_____|_____|_____|_____| \n");
     } else {
        current = head;
        while(current != NULL) {
           sprintf(price_buffer, "$%.2f", current->price);
           printf("|                                         |            |          |         | \n");
           printf("| %-30s | %8.1fGB | %6dh   | %7s | \n", current->name, current->space, current->playtime, price_buffer);
           printf("|_____|_____|_____|_____| \n");
           current = current->next;
        }
     }
     printf("\n\n");
}


int main(void) {
   char selection[5] = "";   // add or delete

   printf("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\
   printf("This program take videogames' names, required space, playtime, and price as input. \n"
          "It then puts them in a liked list database. \n\n"
          "Press enter to continue \n");
   getchar();

   while(strcmp(selection, "done") != 0 && strcmp(selection, "Done") != 0 && strcmp(selection, "DONE") != 0) {
      print_nodes();

        add_node();
        print_nodes();
        printf("Would you like to continue? \nType y for yes or done to quit. \n\n?..");
        fgets(selection, 5, stdin);
     }

     return 0;
}
```

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 (1 ENTRIES):

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

Enter name (will read up to 30 characters): Minecraft
Enter required space in GB (up to 9,999.9): 0.6
Enter playtime in hours (up to 99,999): 200
Enter price in USD (enter up to 999.99): 23.99█

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	110.0GB	1126h	$10.99
Rainbow Six Siege	82.3GB	1312h	$6.89
Garfield Kart	1.8GB	12h	$14.99

Would you like to continue?
Type y for yes or done to quit.

?..done