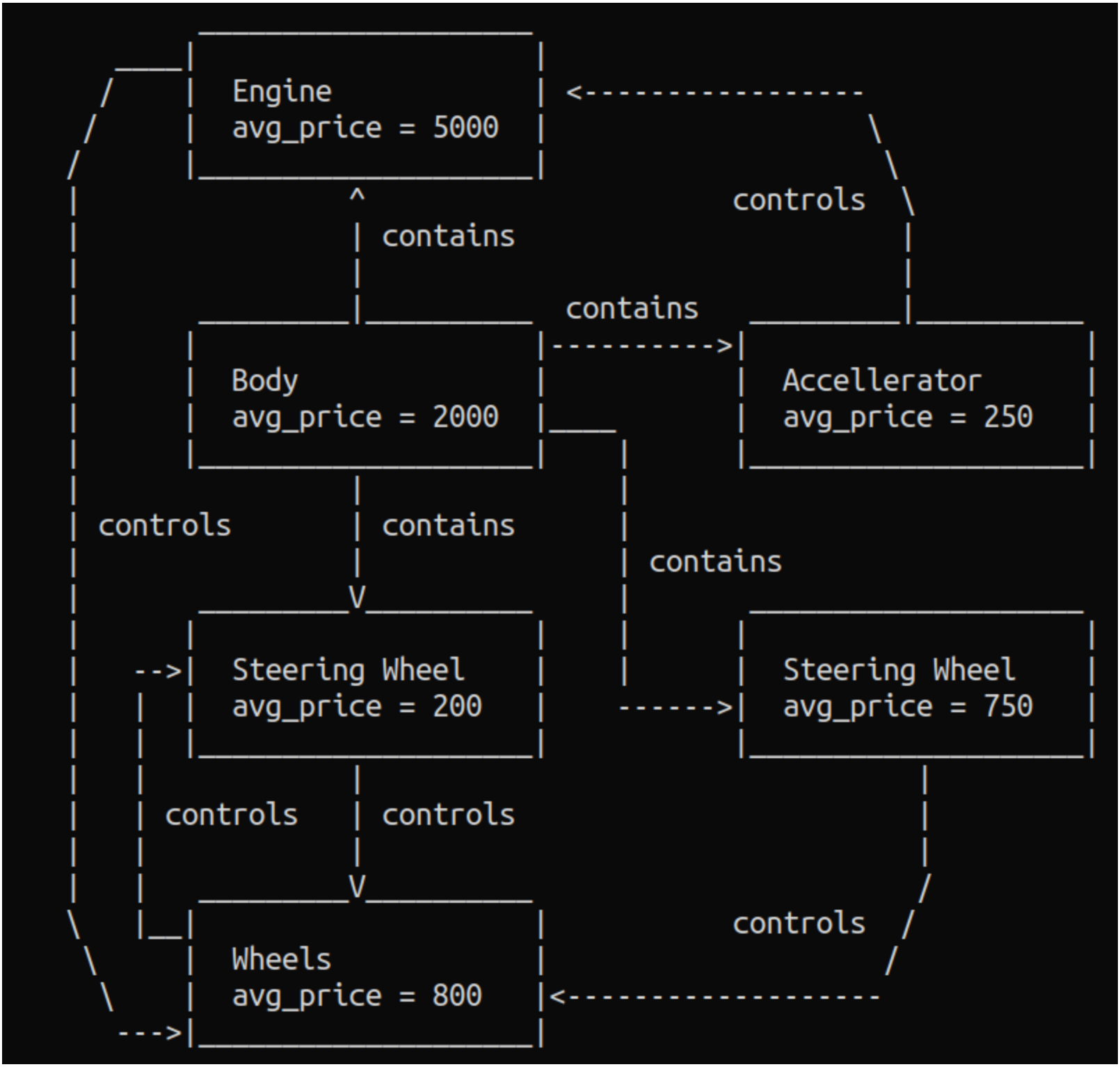

Engine
avg_price = 5000
contains
controls
Body
avg_price = 2000
contains
Accellerator
avg_price = 250
controls
contains
contains
Steering Wheel
avg_price = 200
Steering Wheel
avg_price = 750
controls
controls
controls
Wheels
avg_price = 800

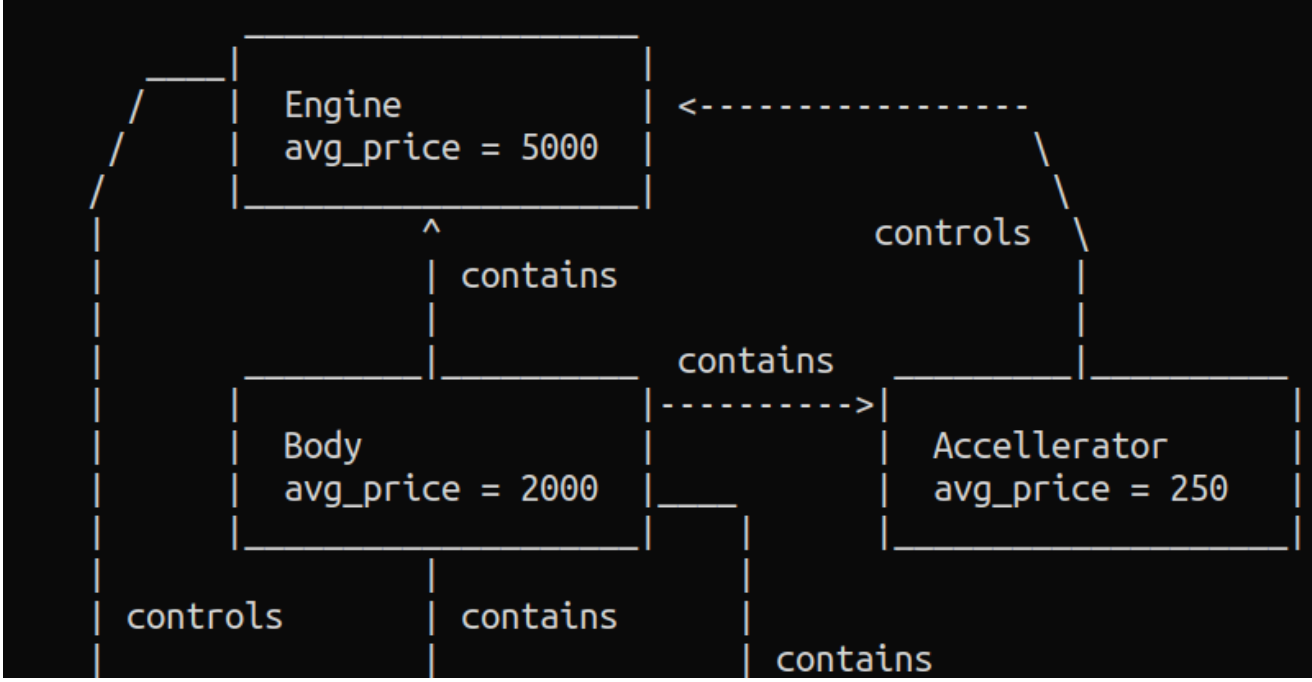
```c
/*
 * C program by Dave Russillo. Made for CS1311.
 * Representation of car parts and their relationships using a network/graph.
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>


struct Part {
  char name[80];
  int avg_price;  // in USD
  struct Part **contains;  // pointer to array of pointers
  struct Part *controls;
};


enum Mode {
  NORMAL,
  DEBUG
} mode;


void populate_network(struct Part *start) {
  void populate_network_helper(struct Part *node,
                               char *name,
                               int avg_price,
                               int contains_amount,
                               struct Part *controls) {
    // assign name
    strcpy(node->name, name);

    // assign average price
    node->avg_price = avg_price;

    // initialize contains array
    struct Part **contains = malloc(sizeof(struct Part*) * contains_amount);
    for(int i = 0; i < contains_amount; i++) {
      // initialize each element
      contains[i] = malloc(sizeof(struct Part));
    }
    // assign contains
    node->contains = contains;

    // assign controls
    node->controls = controls;

    if(mode == DEBUG) {
      printf("New part added---name: %s---location: %p\n", node->name, node);
    }
  }
  // Body: Contains Engine, Accellerator, Braking System, and Steering Wheel. Controls nothing.
  populate_network_helper(start, "Body", 2000, 4, NULL);
  // Engine: Contains nothing. Controls Wheels.
  populate_network_helper(start->contains[0], "Engine", 5000, 0, (struct Part*)malloc(sizeof(struct Part)));
  // Wheels: Contains nothing. Controls Body.
  populate_network_helper(start->contains[0]->controls, "Wheels", 800, 0, start);
  // Accellerator: Contains nothing. Controls Engine.
  populate_network_helper(start->contains[1], "Accellerator", 250, 0, start->contains[0]);
  // Braking System: Contains nothing. Controls Wheels.
  populate_network_helper(start->contains[2], "Braking System", 750, 0, start->contains[0]->controls);
```

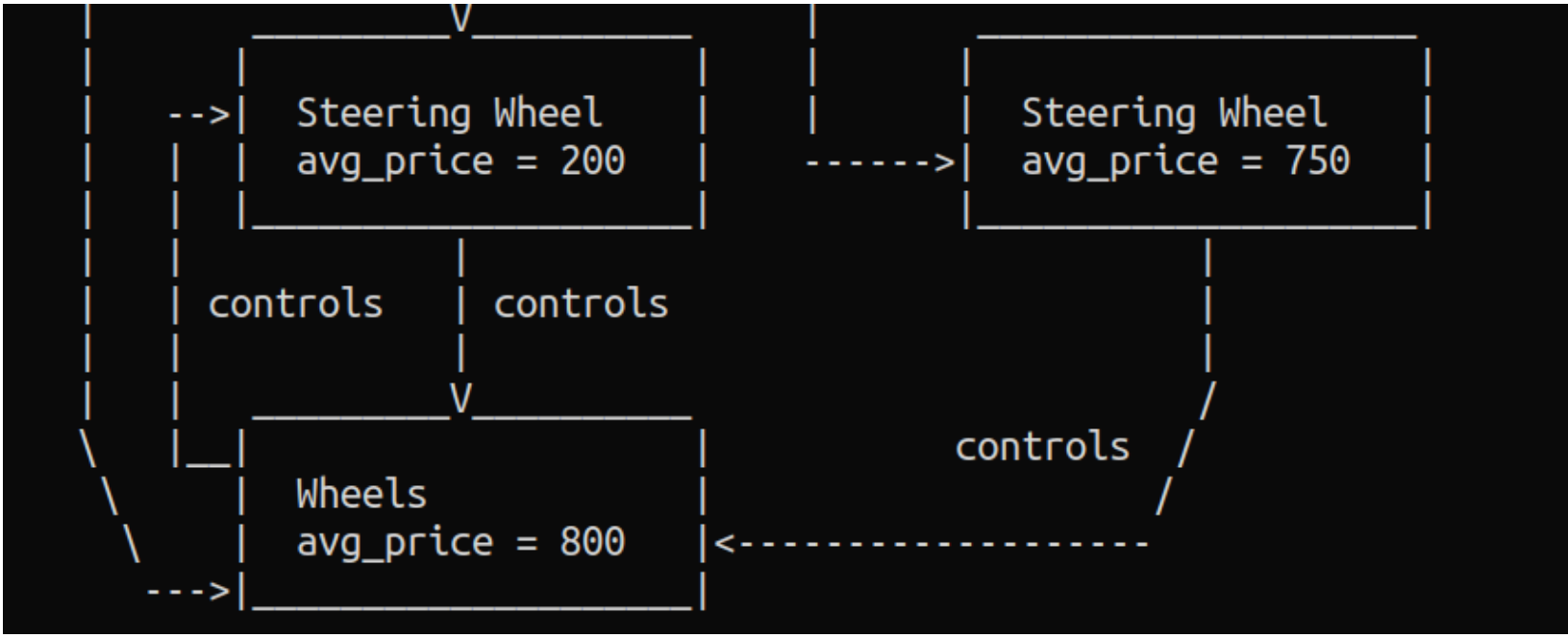
```
  // Steering Wheel: Contains nothing. Controls Wheels.
  populate_network_helper(start->contains[3], "Steering Wheel", 200, 0, start->contains[0]->controls);

  // check if all nodes exist
  if(mode == DEBUG) {
    if(start != NULL &&  // Body
       start->contains[0] != NULL &&  // Engine
       start->contains[1] != NULL &&  // Accellerator
       start->contains[2] != NULL &&  // Braking System
       start->contains[3] != NULL &&  // Steering Wheel
       start->contains[3]->controls != NULL) {  // Wheels
      printf("All Nodes successfully populated\n");
    } else{
      printf("ERROR: One or more Nodes were not successfully initialized\n");
      exit(1);
    }
    printf("\n");
  }
}


void print_network(struct Part *start) {
  void print_network_helper(struct Part *node) {
    // name and price
    printf("The part named '%s' has an average price of $%d", node->name, node->avg_price);
    if(mode == DEBUG) {
      printf(" and it is located at %p", node);
    }
    printf(".\n");

    // contains
```

```
94      if(node->contains[0] != NULL) {
95        printf("    It contains the following parts:\n");
96        for(int i = 0; node->contains[i] != NULL; i++) {
97          printf("        - %s", node->contains[i]->name);
98          if(mode == DEBUG) {
99            printf(" (at %p)", node->contains[i]);
100         }
101         printf("\n");
102       }
103     }
104
105     // controls
106     if(node->controls != NULL) {
107       printf("    It controls the part '%s'", node->controls->name);
108       if(mode == DEBUG) {
109         printf(" (located at %p)", node->controls);
110       }
111       printf(".\n");
112     }
113     printf("\n");
114   }
115   // call on each node
116   print_network_helper(start);  // Body
117   print_network_helper(start->contains[0]);  // Engine
118   print_network_helper(start->contains[1]);  // Accellerator
119   print_network_helper(start->contains[2]);  // Braking System
120   print_network_helper(start->contains[3]);  // Steering Wheel
121   print_network_helper(start->contains[3]->controls);  // Wheels
122 }
123
124
```

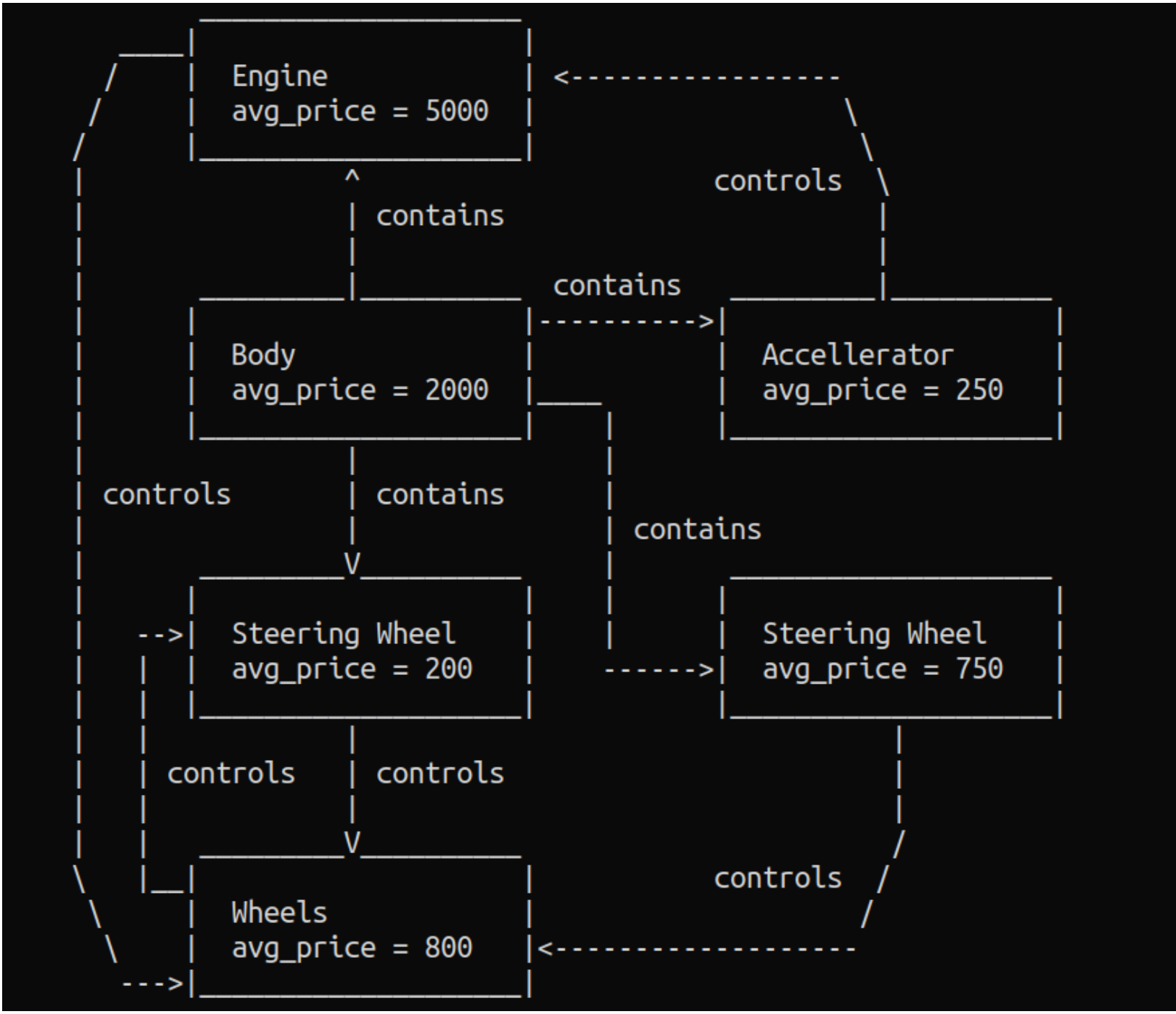
```c
void print_network_ascii(void) {
  printf(
         "\n----------ASCII Representation of network----------\n"
         "                  ____________________\n"
         "             ____|                    |\n"
         "          /      |  Engine            | <-----------------\n"
         "        /        |  avg_price = 5000  |                    \\\\\n"
         "      /          |____________________|                     \\\\\n"
         "     |                    ^                          controls  \\\\\n"
         "     |                    | contains                             |\n"
         "     |                    |                                      |\n"
         "     |          __________|__________  contains   __________|__________\n"
         "     |          |                     |---------->|                     |\n"
         "     |          |  Body               |           |  Accellerator       |\n"
         "     |          |  avg_price = 2000   |____       |  avg_price = 250    |\n"
         "     |          |_____________________|    |      |_____________________|\n"
         "     |                    |                    |\n"
         "     | controls           | contains           |\n"
         "     |                    |                    | contains\n"
         "     |          __________V__________          |          ____________________\n"
         "     |          |                     |         |          |                   |\n"
         "     |      -->|  Steering Wheel     |         |          |  Steering Wheel   |\n"
         "     |      |   |  avg_price = 200    |         ------->|  avg_price = 750  |\n"
         "     |      |   |_____________________|                   |___________________|\n"
         "     |      |              |                                          |\n"
         "     |      | controls     | controls                                 |\n"
         "     |      |              |                                          |\n"
         "     |      |   __________V__________                                /\n"
         "     \\      |__|                     |                 controls    /\n"
         "      \\         |  Wheels             |                           /\n"
         "       \\        |  avg_price = 800    |<--------------------------\n"
         "        --->|_____________________|\n\n");
}


int main(int argc, char *argv[]) {
  if(argc == 2 && strcmp(argv[1], "debug") == 0) {
    mode = DEBUG;
    puts("-----DEBUG MODE-----");
  } else if(argc == 1) {
    mode = NORMAL;
  } else {
    puts("Invalid arguments.");
  }

  struct Part *start = malloc(sizeof(struct Part)); // declare and initialize start
  populate_network(start);
  print_network(start);
  print_network_ascii();

  return 0;
}
```

```
The part named 'Body' has an average price of $2000.
    It contains the following parts:
        - Engine
        - Accellerator
        - Braking System
        - Steering Wheel

The part named 'Engine' has an average price of $5000.
    It controls the part 'Wheels'.

The part named 'Accellerator' has an average price of $250.
    It controls the part 'Engine'.

The part named 'Braking System' has an average price of $750.
    It controls the part 'Wheels'.

The part named 'Steering Wheel' has an average price of $200.
    It controls the part 'Wheels'.

The part named 'Wheels' has an average price of $800.
    It controls the part 'Body'.


----------ASCII Representation of network----------
```




Steering Wheel
avg_price = 200
Steering Wheel
avg_price = 750
controls
controls
controls
Wheels
avg_price = 800

```
-----DEBUG MODE-----
New part added---name: Body---location: 0x5d135a2836b0
New part added---name: Engine---location: 0x5d135a283750
New part added---name: Wheels---location: 0x5d135a283910
New part added---name: Accellerator---location: 0x5d135a2837c0
New part added---name: Braking System---location: 0x5d135a283830
New part added---name: Steering Wheel---location: 0x5d135a2838a0
All Nodes successfully populated

The part named 'Body' has an average price of $2000 and it is located at 0x5d135a2836b0.
    It contains the following parts:
        - Engine (at 0x5d135a283750)
        - Accellerator (at 0x5d135a2837c0)
        - Braking System (at 0x5d135a283830)
        - Steering Wheel (at 0x5d135a2838a0)

The part named 'Engine' has an average price of $5000 and it is located at 0x5d135a283750.
    It controls the part 'Wheels' (located at 0x5d135a283910).

The part named 'Accellerator' has an average price of $250 and it is located at 0x5d135a2837c0.
    It controls the part 'Engine' (located at 0x5d135a283750).

The part named 'Braking System' has an average price of $750 and it is located at 0x5d135a283830.
    It controls the part 'Wheels' (located at 0x5d135a283910).

The part named 'Steering Wheel' has an average price of $200 and it is located at 0x5d135a2838a0.
    It controls the part 'Wheels' (located at 0x5d135a283910).

The part named 'Wheels' has an average price of $800 and it is located at 0x5d135a283910.
    It controls the part 'Body' (located at 0x5d135a2836b0).


----------ASCII Representation of network----------
```


Engine
avg_price = 5000
contains
controls
contains
Body
avg_price = 2000
contains
Accellerator
avg_price = 250
controls
contains
contains
Steering Wheel
avg_price = 200
Steering Wheel
avg_price = 750
controls
controls
controls
Wheels
avg_price = 800