```c
/* BY SUBMITTING THIS FILE TO CARMEN, I CERTIFY THAT I HAVE PERFORMED ALL OF
THE WORK TO CREATE THIS FILE AND/OR DETERMINE THE ANSWERS FOUND WITHIN
THIS FILE MYSELF WITH NO ASSISSTANCE FROM ANY PERSON (OTHER THAN THE
INSTRUCTOR OR GRADERS OF THIS COURSE) AND I HAVE STRICTLY ADHERED TO THE
TENURES OF THE OHIO STATE UNIVERSITY'S ACADEMIC INTEGRITY POLICY.
*/
#include "lab4.h"
/*Adds a new student*/
void option8(Node* head, char *categories) {
        int id; /*Temporary variable to store student ID number for new student*/
        int count; /*Number of scores for the new student in each category*/
        float cumulative; /*Temporary variable to store the cumulative score for each
category*/
        Node* newNodePtr; /*The new node for the new student*/
        float grade; /*Temporary variable to store the grade*/
        struct Data student; /*Structure to store the data for the new student*/
        /*Prompts the user for the name*/
        printf("Enter the Student's Name: ");
        scanf(" %[^\n]", student.student_name);
        /*Prompts the user for the ID number*/
        printf("Enter the Student's ID Number: ");
        scanf("%d", &id);
        /*If the ID is duplicate, prompts the user to enter another one until it is
no longer a duplicate*/
        while (ID_isduplicate(head, id)) {
                printf("Student ID Number entered was a duplicate.\n");
                printf("Enter the Student's ID Number: ");
                scanf("%d", &id);
        }
        student.student_ID = id;
        /*Prompt the individual scores for the new student with -1 as no score*/
        printf("Enter first %s score (use -1 if there is no score): ",
(char*)categories);
        scanf("%f", &student.cat1.score1);
        printf("Enter second %s score (use -1 if there is no score): ",
(char*)categories);
        scanf("%f", &student.cat1.score2);
        printf("Enter third %s score (use -1 if there is no score): ",
(char*)categories);
        scanf("%f", &student.cat1.score3);
        printf("Enter first %s score (use -1 if there is no score): ",
(char*)categories+15);
        scanf("%f", &student.cat2.score1);
        printf("Enter second %s score (use -1 if there is no score): ",
(char*)categories+15);
        scanf("%f", &student.cat2.score2);
        printf("Enter third %s score (use -1 if there is no score): ",
(char*)categories+15);
        scanf("%f", &student.cat2.score3);
        printf("Enter first %s score (use -1 if there is no score): ",
(char*)categories+30);
        scanf("%f", &student.cat3.score1);
        printf("Enter second %s score (use -1 if there is no score): ",
(char*)categories+30);
        scanf("%f", &student.cat3.score2);
        printf("Enter third %s score (use -1 if there is no score): ",
(char*)categories+30);
        scanf("%f", &student.cat3.score3);
        printf("Enter first %s score (use -1 if there is no score): ",
(char*)categories+45);
        scanf("%f", &student.cat4.score1);
        printf("Enter second %s score (use -1 if there is no score): ",
```

```
(char*)categories+45);
        scanf("%f", &student.cat4.score2);
        printf("Enter third %s score (use -1 if there is no score): ",
(char*)categories+45);
        scanf("%f", &student.cat4.score3);
        /*Calculates the cumulative scores for each category for the new student*/
        /*Category 1*/
        count = 0;
        cumulative = 0;
        if (student.cat1.score1 != -1) {
                cumulative += student.cat1.score1;
                count++;
        }
        if (student.cat1.score2 != -1) {
                cumulative += student.cat1.score2;
                count++;
        }
        if (student.cat1.score3 != -1) {
                cumulative += student.cat1.score3;
                count++;
        }
        if (count != 0) {
                cumulative /= count;
        }
        else {
                cumulative = -1;
        }
        student.cat1.cumulative = cumulative;
        /*Category 2*/
        count = 0;
        cumulative = 0;
        if (student.cat2.score1 != -1) {
                cumulative += student.cat2.score1;
                count++;
        }
        if (student.cat2.score2 != -1) {
                cumulative += student.cat2.score2;
                count++;
        }
        if (student.cat2.score3 != -1) {
                cumulative += student.cat2.score3;
                count++;
        }
        if (count != 0) {
                cumulative /= count;
        }
        else {
                cumulative = -1;
        }
        student.cat2.cumulative = cumulative;
        /*Category 3*/
        cumulative = 0;
        count = 0;
        if (student.cat3.score1 != -1) {
                cumulative += student.cat3.score1;
                count++;
        }
        if (student.cat3.score2 != -1) {
                cumulative += student.cat3.score2;
                count++;
        }
        if (student.cat3.score3 != -1) {
```

```
                cumulative += student.cat3.score3;
                count++;
        }
        if (count != 0) {
                cumulative /= count;
        }
        else {
                cumulative = -1;
        }
        student.cat3.cumulative = cumulative;
        count = 0;
        cumulative = 0;
        /*Category 4*/
        if (student.cat4.score1 != -1) {
                cumulative += student.cat4.score1;
                count++;
        }
        if (student.cat4.score2 != -1) {
                cumulative += student.cat4.score2;
                count++;
        }
        if (student.cat4.score3 != -1) {
                cumulative += student.cat4.score3;
                count++;
        }
        if (count != 0) {
                cumulative /= count;
        }
        else {
                cumulative = -1;
        }
        student.cat4.cumulative = cumulative;
        /*Calculates the current grade for the new student. Use a calculation with 0
for each score with -1.*/
        grade = 0;
        if (student.cat1.cumulative != -1) {
                grade += 0.15 * student.cat1.cumulative;
        }
        if (student.cat2.cumulative != -1) {
                grade += 0.3 * student.cat2.cumulative;
        }
        if (student.cat3.cumulative != -1) {
                grade += 0.2 * student.cat3.cumulative;
        }
        if (student.cat4.cumulative != -1) {
                grade += 0.35 * student.cat4.cumulative;
        }
        student.current_grade = grade;
        student.final_grade = -1;
        /*Allocates a node for the new student and then inserts it*/
        newNodePtr = allocate_node(student);
        head = insertNode(head, newNodePtr);
        /*Confirms the user that the new student has been added and then prints the
data*/
        printf("%s, Student ID# %d has been added with the following information:\n",
student.student_name, student.student_ID);
        printHeader(categories);
        printStudent(newNodePtr);
}
```