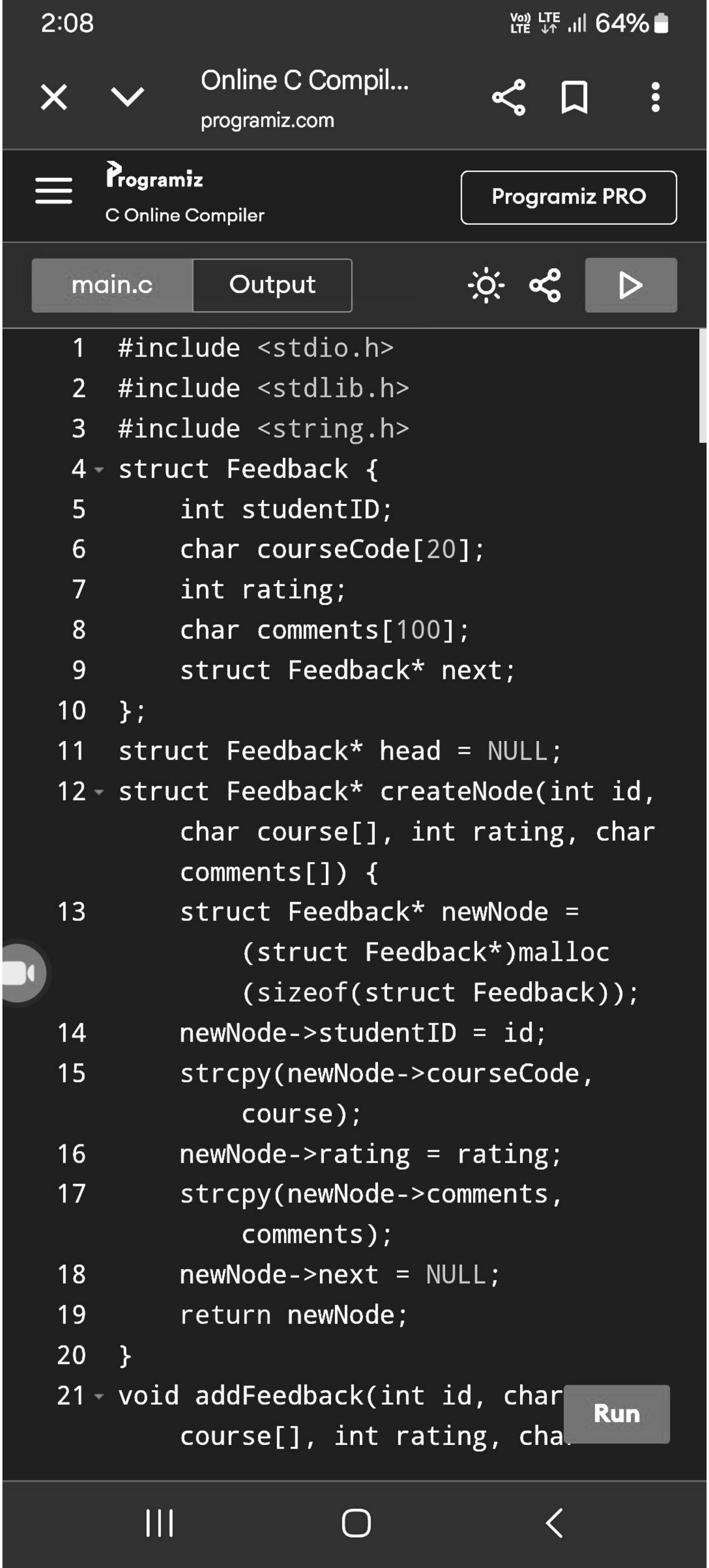
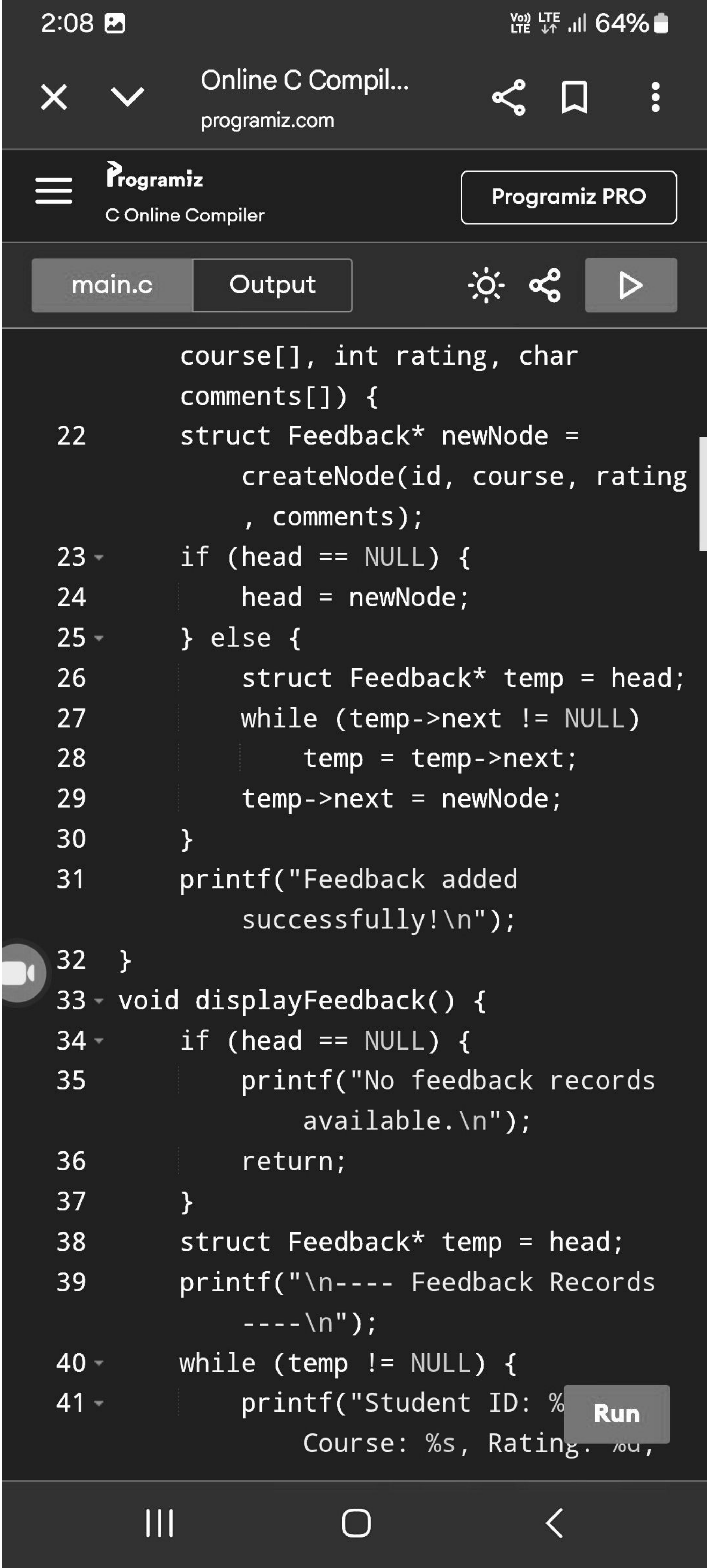
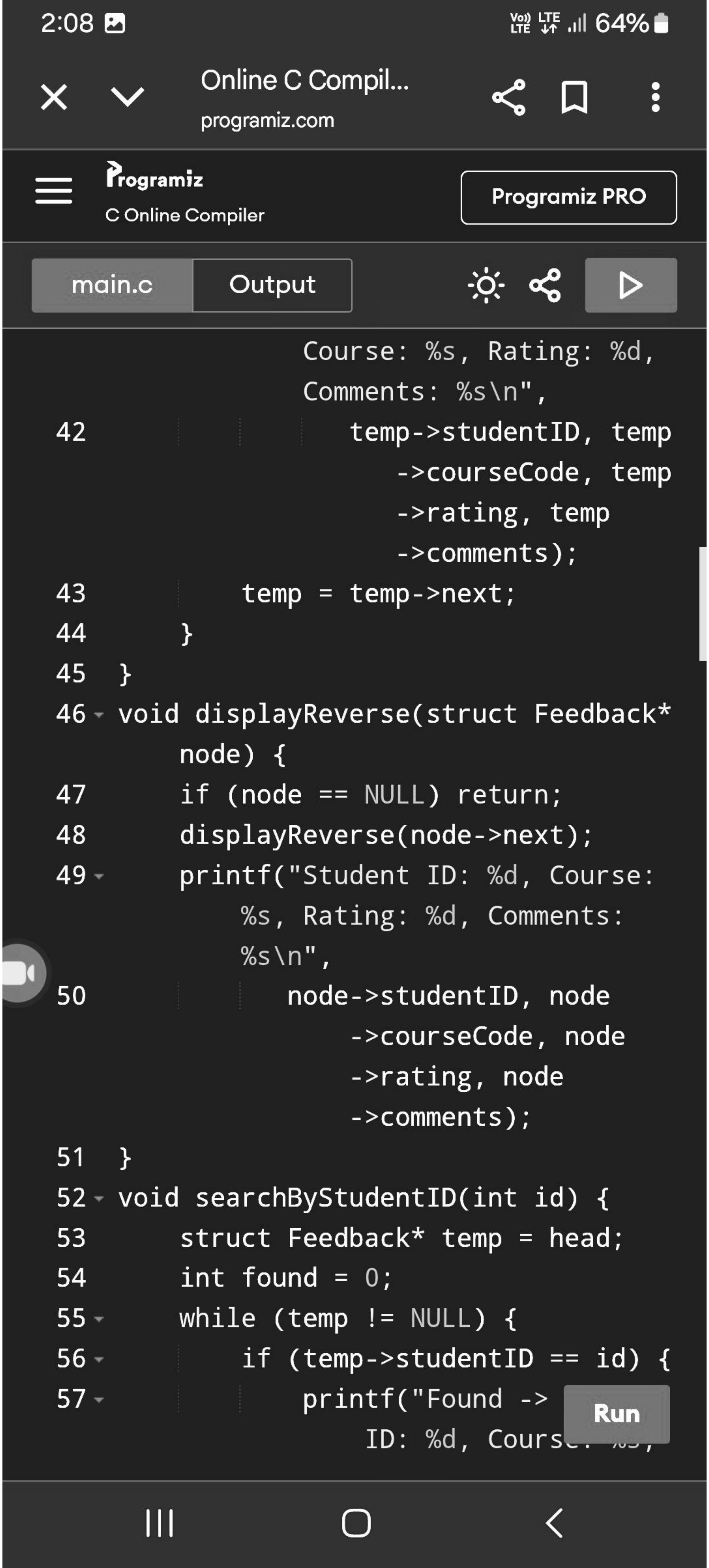


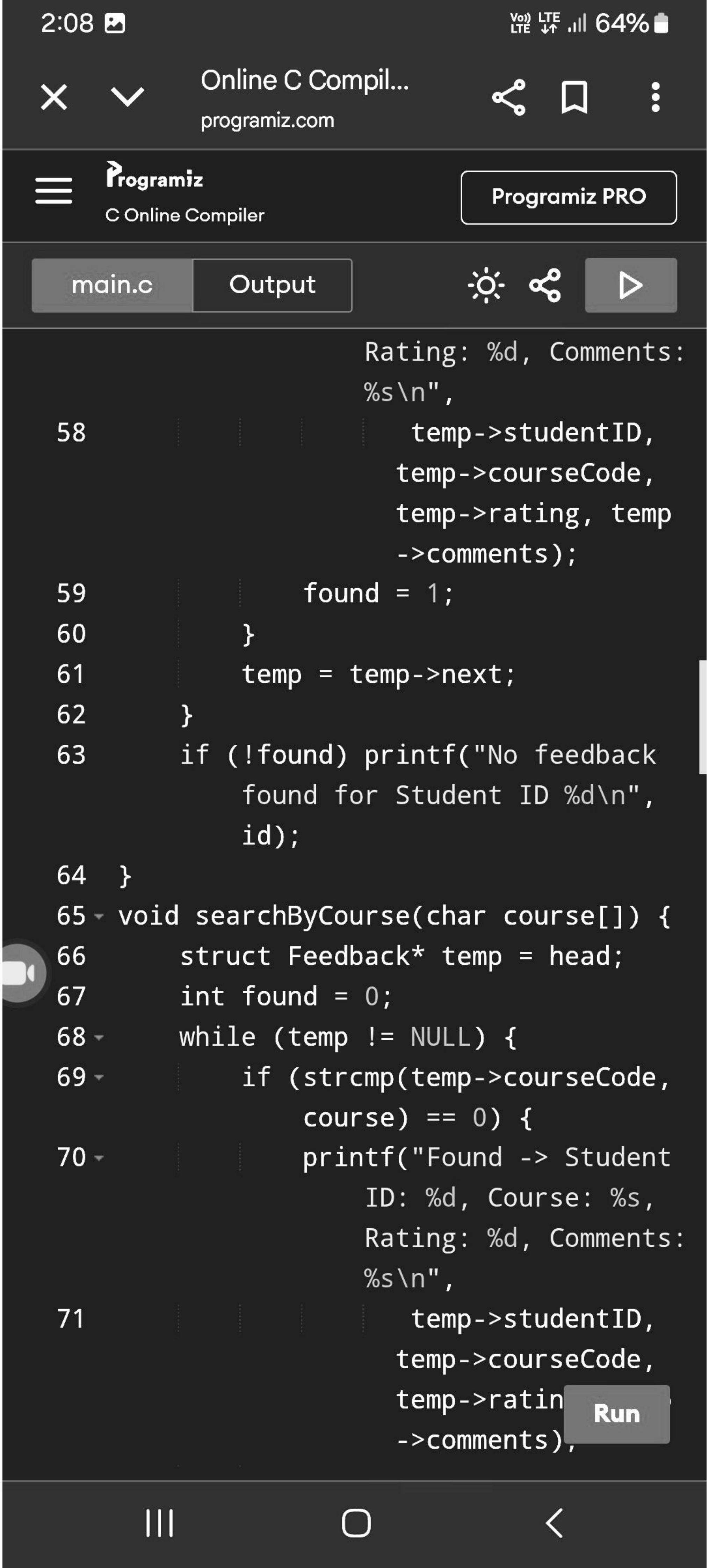
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Feedback {
    int studentID;
    char courseCode[20];
    int rating;
    char comments[100];
    struct Feedback* next;
};
struct Feedback* head = NULL;
struct Feedback* createNode(int id, char course[], int rating, char comments[]) {
    struct Feedback* newNode = (struct Feedback*)malloc(sizeof(struct Feedback));
    newNode->studentID = id;
    strcpy(newNode->courseCode, course);
    newNode->rating = rating;
    strcpy(newNode->comments, comments);
    newNode->next = NULL;
    return newNode;
}
void addFeedback(int id, char course[], int rating, cha
```



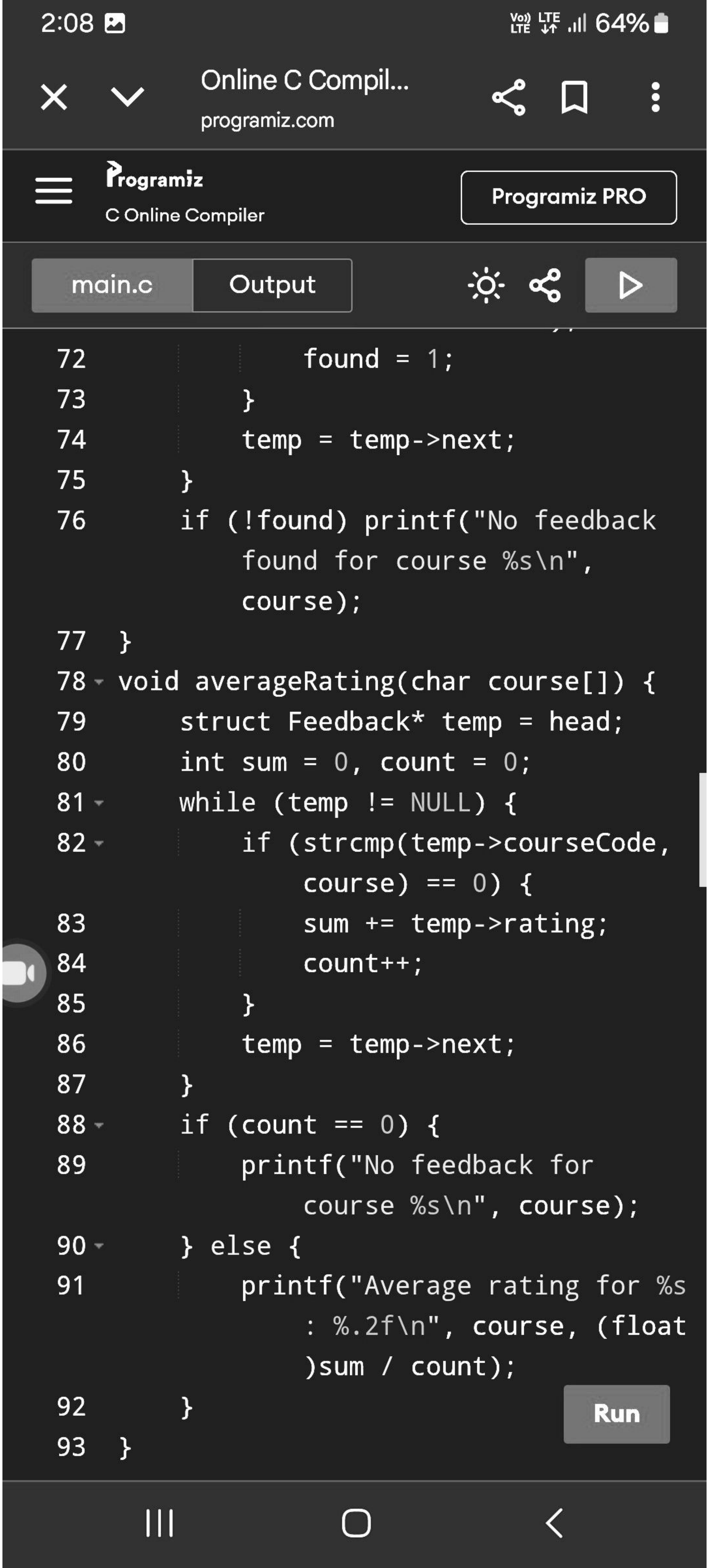
```c
course[], int rating, char comments[]) {
    struct Feedback* newNode = createNode(id, course, rating, comments);
    if (head == NULL) {
        head = newNode;
    } else {
        struct Feedback* temp = head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newNode;
    }
    printf("Feedback added successfully!\n");
}
void displayFeedback() {
    if (head == NULL) {
        printf("No feedback records available.\n");
        return;
    }
    struct Feedback* temp = head;
    printf("\n---- Feedback Records ----\n");
    while (temp != NULL) {
        printf("Student ID: % Course: %s, Rating: %d,
```



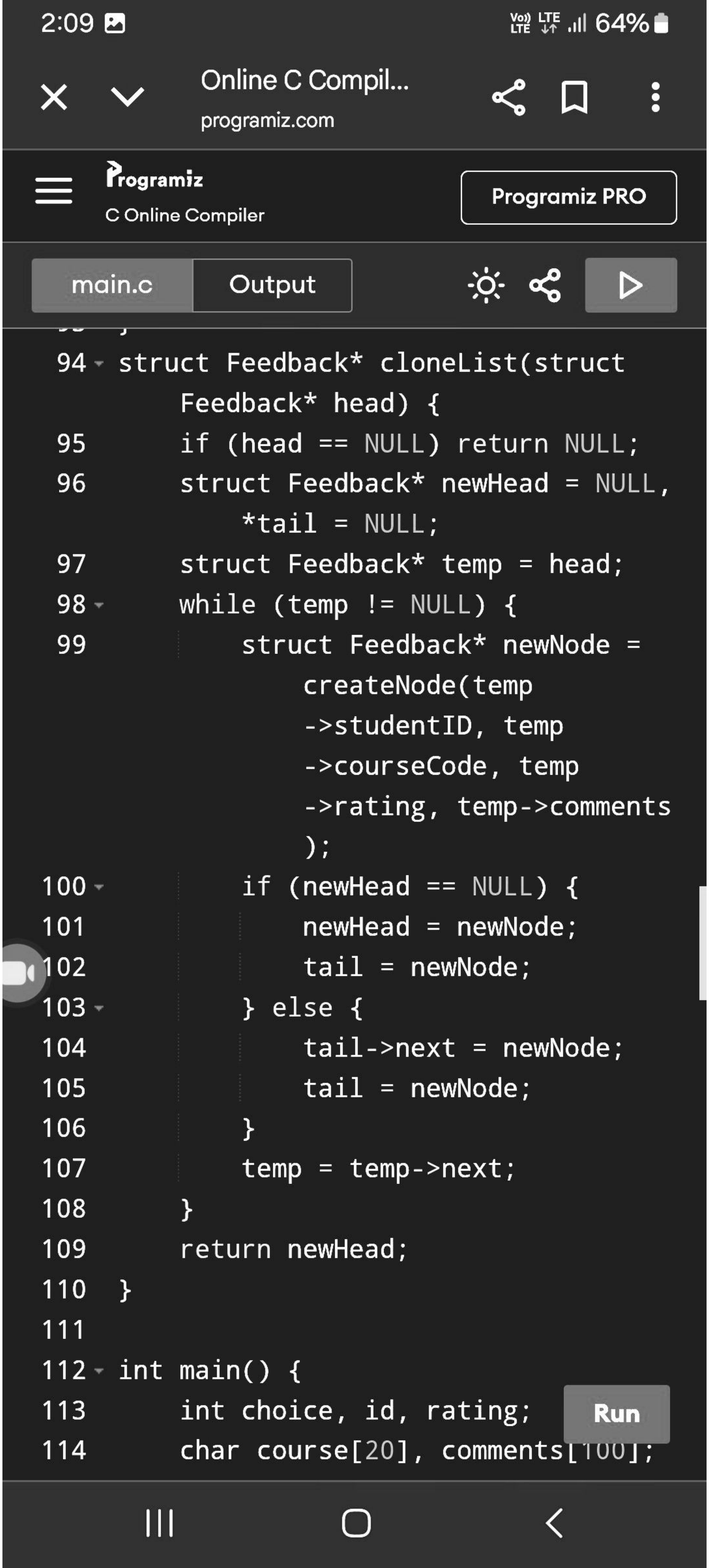
```
                Course: %s, Rating: %d, Comments: %s\n",
                temp->studentID, temp->courseCode, temp->rating, temp->comments);
            temp = temp->next;
        }
}
void displayReverse(struct Feedback* node) {
    if (node == NULL) return;
    displayReverse(node->next);
    printf("Student ID: %d, Course: %s, Rating: %d, Comments: %s\n",
            node->studentID, node->courseCode, node->rating, node->comments);
}
void searchByStudentID(int id) {
    struct Feedback* temp = head;
    int found = 0;
    while (temp != NULL) {
        if (temp->studentID == id) {
            printf("Found -> ID: %d, Cours
```



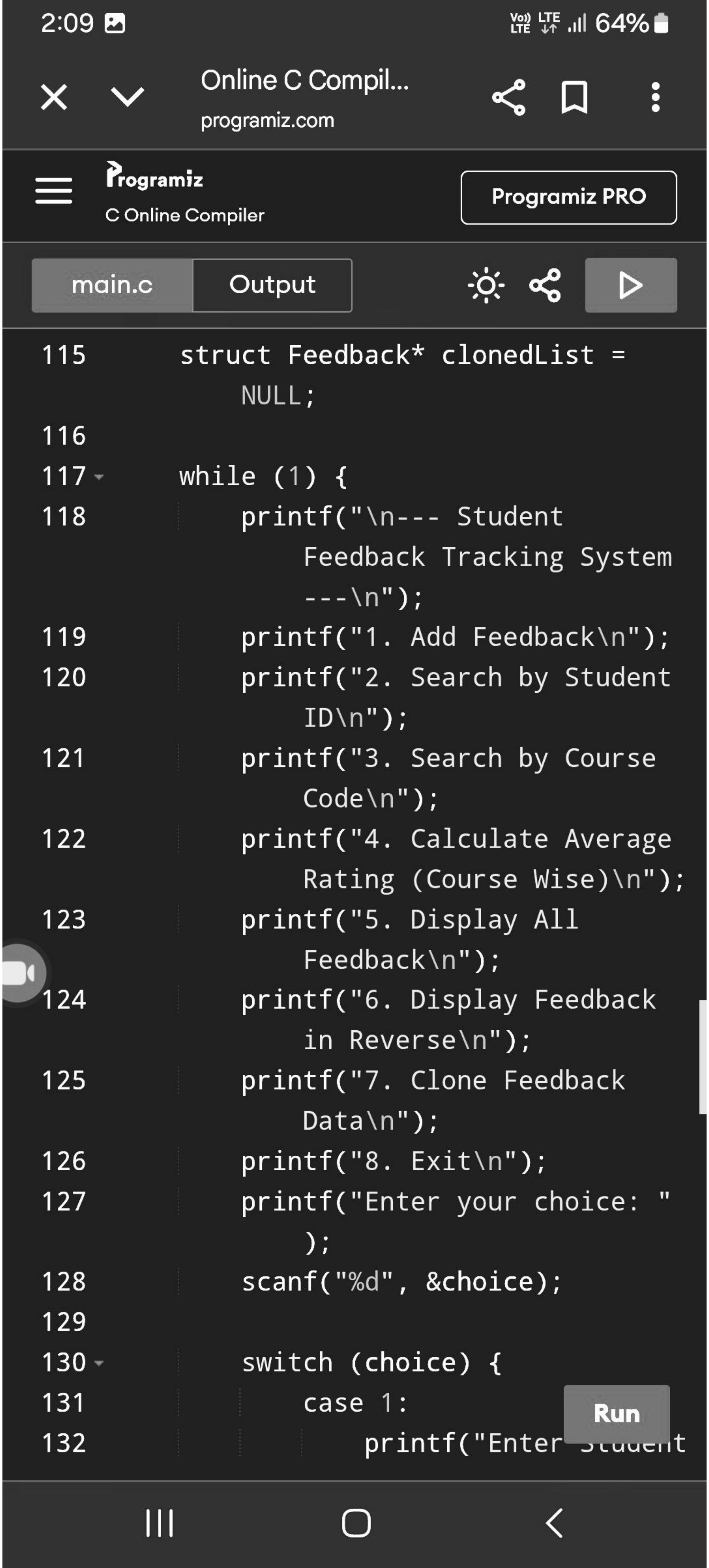
```
                Rating: %d, Comments: %s\n",
                    temp->studentID, temp->courseCode, temp->rating, temp->comments);
            found = 1;
        }
        temp = temp->next;
    }
    if (!found) printf("No feedback found for Student ID %d\n", id);
}
void searchByCourse(char course[]) {
    struct Feedback* temp = head;
    int found = 0;
    while (temp != NULL) {
        if (strcmp(temp->courseCode, course) == 0) {
            printf("Found -> Student ID: %d, Course: %s, Rating: %d, Comments: %s\n",
                    temp->studentID, temp->courseCode, temp->ratin
                    ->comments);
```



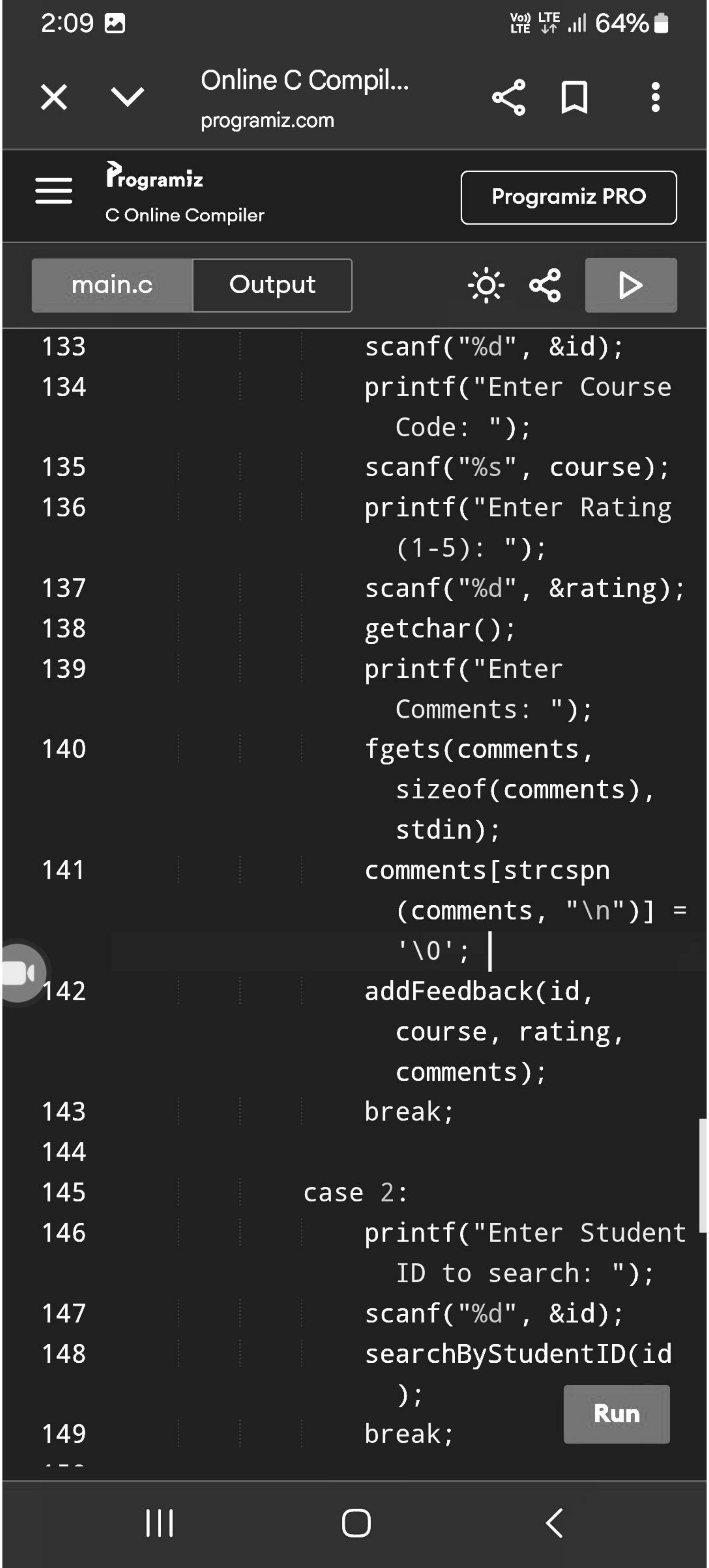
```c
            found = 1;
        }
        temp = temp->next;
    }
    if (!found) printf("No feedback found for course %s\n", course);
}
void averageRating(char course[]) {
    struct Feedback* temp = head;
    int sum = 0, count = 0;
    while (temp != NULL) {
        if (strcmp(temp->courseCode, course) == 0) {
            sum += temp->rating;
            count++;
        }
        temp = temp->next;
    }
    if (count == 0) {
        printf("No feedback for course %s\n", course);
    } else {
        printf("Average rating for %s : %.2f\n", course, (float)sum / count);
    }
}
```



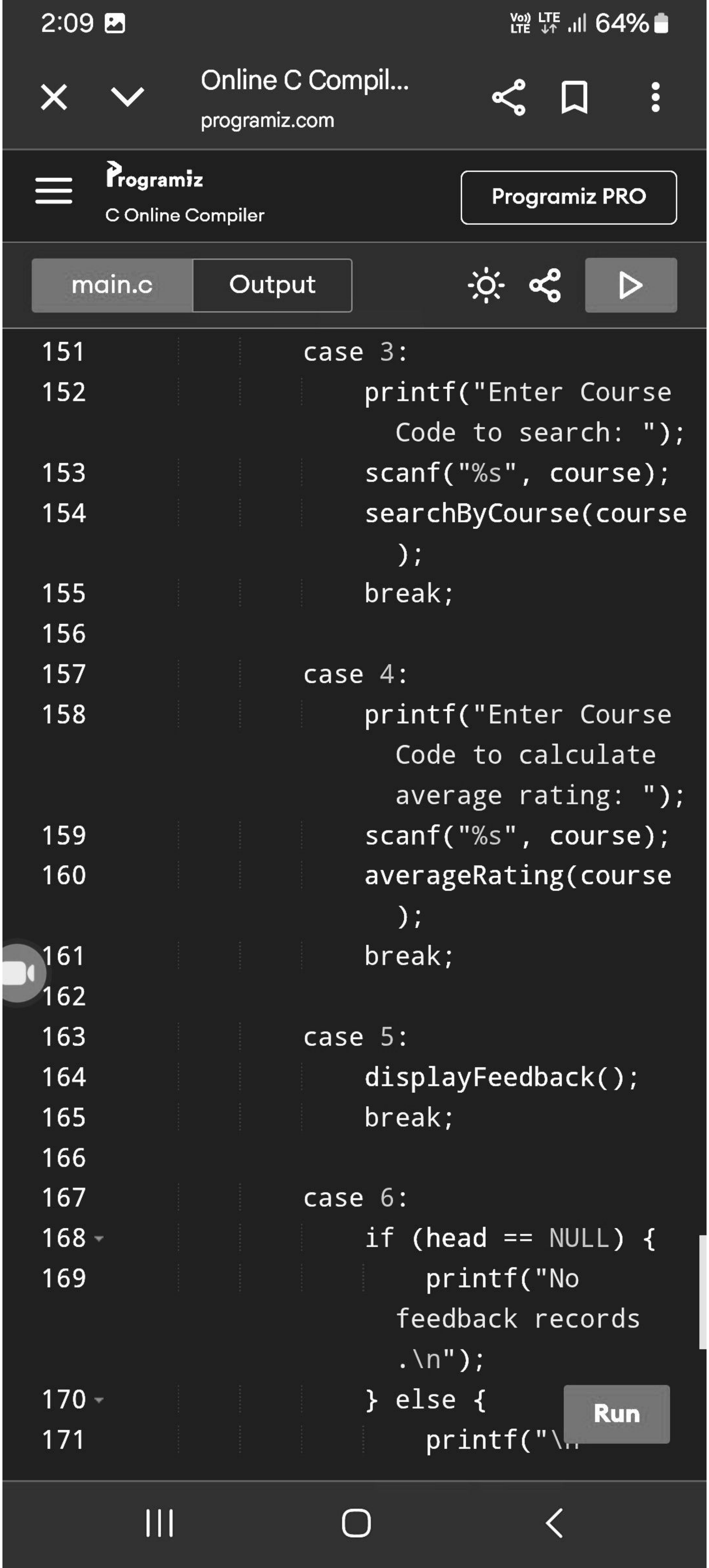
```c
struct Feedback* cloneList(struct Feedback* head) {
    if (head == NULL) return NULL;
    struct Feedback* newHead = NULL, *tail = NULL;
    struct Feedback* temp = head;
    while (temp != NULL) {
        struct Feedback* newNode = createNode(temp->studentID, temp->courseCode, temp->rating, temp->comments);
        if (newHead == NULL) {
            newHead = newNode;
            tail = newNode;
        } else {
            tail->next = newNode;
            tail = newNode;
        }
        temp = temp->next;
    }
    return newHead;
}

int main() {
    int choice, id, rating;
    char course[20], comments[100];
```



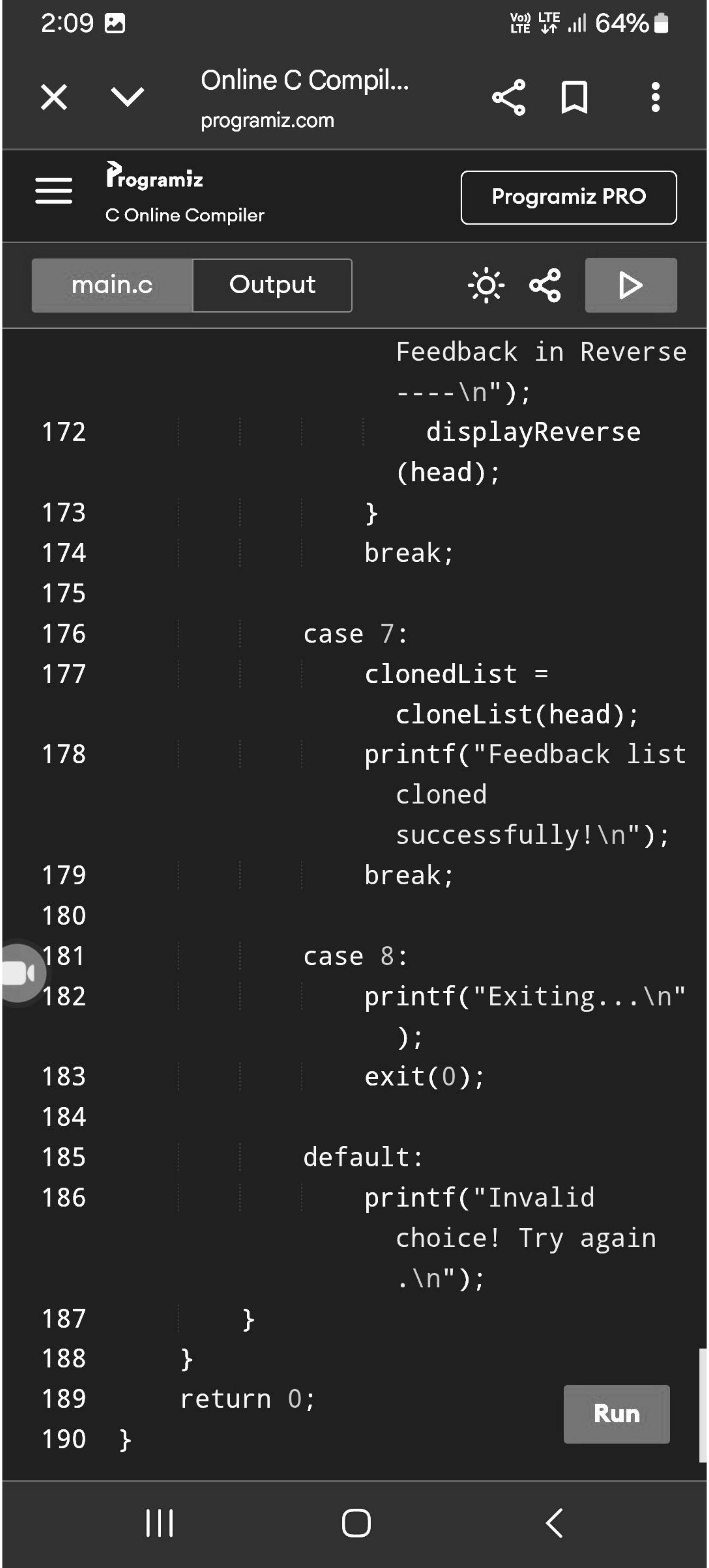
```c
    struct Feedback* clonedList = NULL;

    while (1) {
        printf("\n--- Student Feedback Tracking System ---\n");
        printf("1. Add Feedback\n");
        printf("2. Search by Student ID\n");
        printf("3. Search by Course Code\n");
        printf("4. Calculate Average Rating (Course Wise)\n");
        printf("5. Display All Feedback\n");
        printf("6. Display Feedback in Reverse\n");
        printf("7. Clone Feedback Data\n");
        printf("8. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter
```



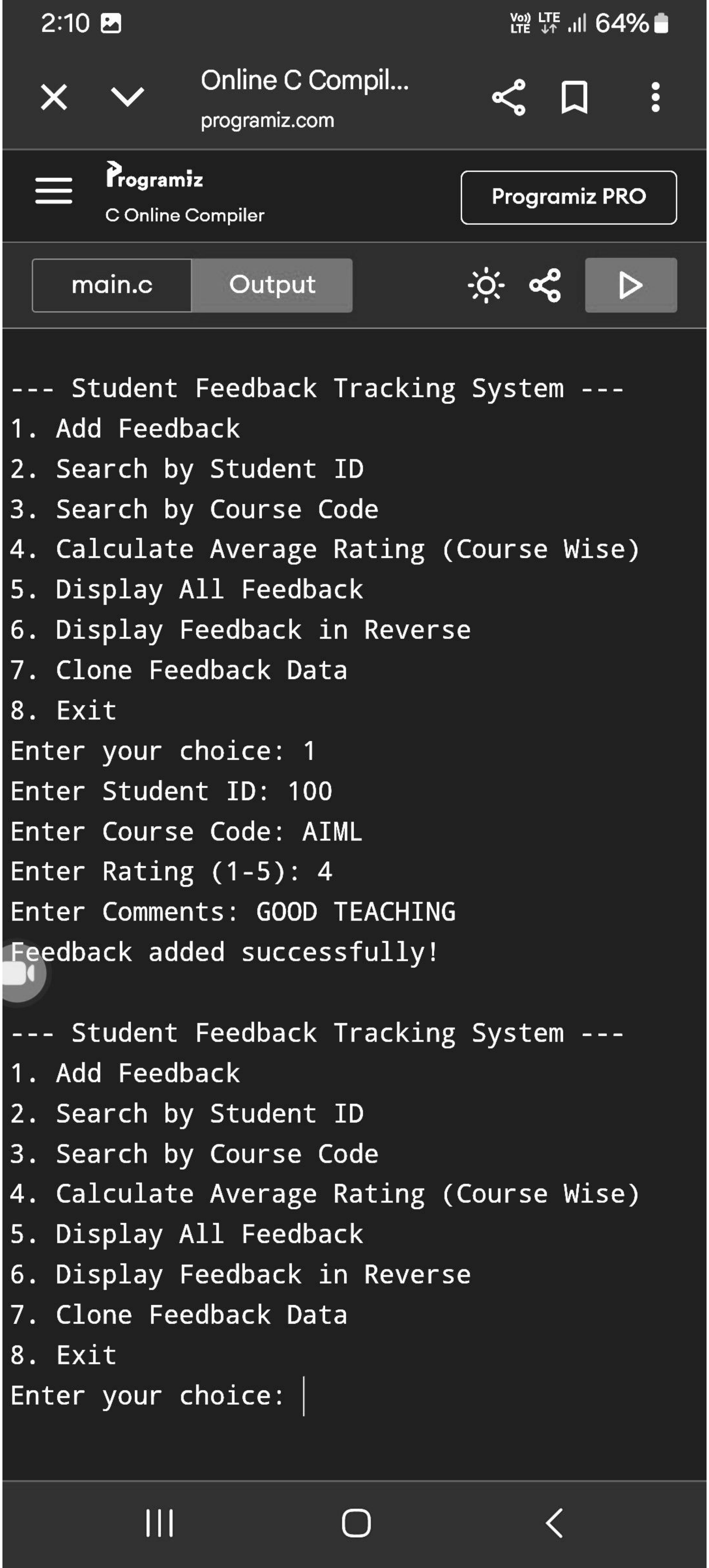
```
                scanf("%d", &id);
                printf("Enter Course Code: ");
                scanf("%s", course);
                printf("Enter Rating (1-5): ");
                scanf("%d", &rating);
                getchar();
                printf("Enter Comments: ");
                fgets(comments, sizeof(comments), stdin);
                comments[strcspn(comments, "\n")] = '\0';
                addFeedback(id, course, rating, comments);
                break;

            case 2:
                printf("Enter Student ID to search: ");
                scanf("%d", &id);
                searchByStudentID(id);
                break;
```



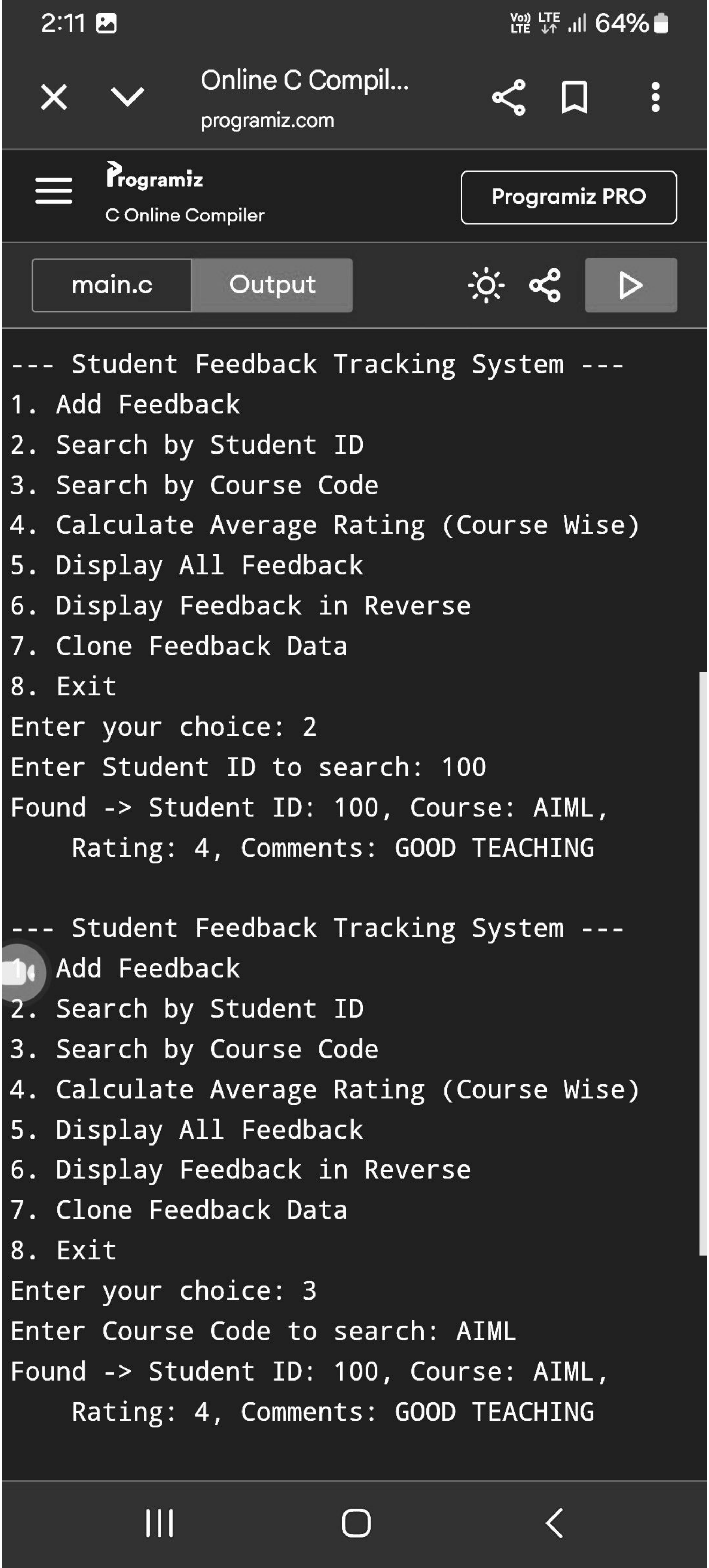
```
        case 3:
            printf("Enter Course Code to search: ");
            scanf("%s", course);
            searchByCourse(course);
            break;

        case 4:
            printf("Enter Course Code to calculate average rating: ");
            scanf("%s", course);
            averageRating(course);
            break;

        case 5:
            displayFeedback();
            break;

        case 6:
            if (head == NULL) {
                printf("No feedback records.\n");
            } else {
                printf("\n
```



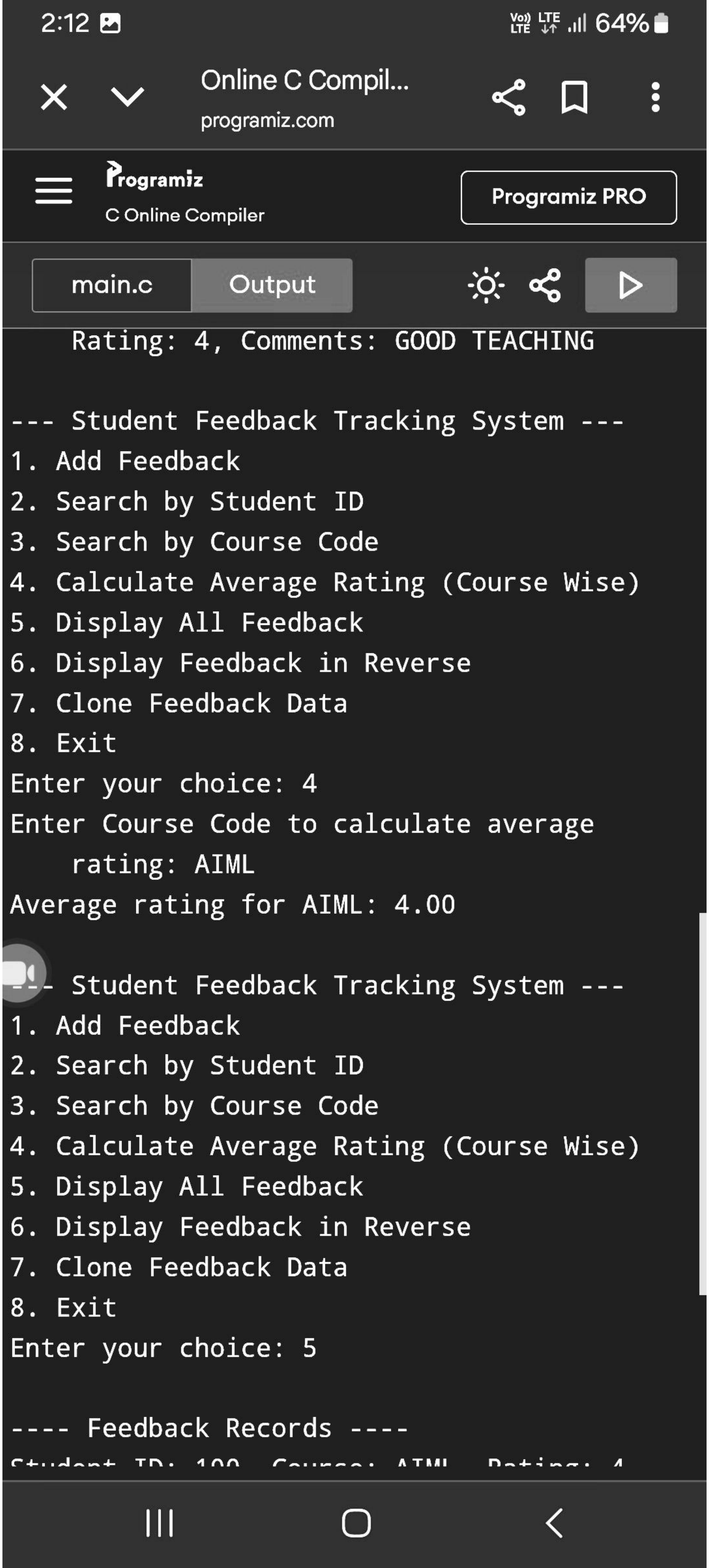
```
                    Feedback in Reverse ----\n");
                    displayReverse(head);
                }
                break;

            case 7:
                clonedList = cloneList(head);
                printf("Feedback list cloned successfully!\n");
                break;

            case 8:
                printf("Exiting...\n");
                exit(0);

            default:
                printf("Invalid choice! Try again.\n");
        }
    }
    return 0;
}
```



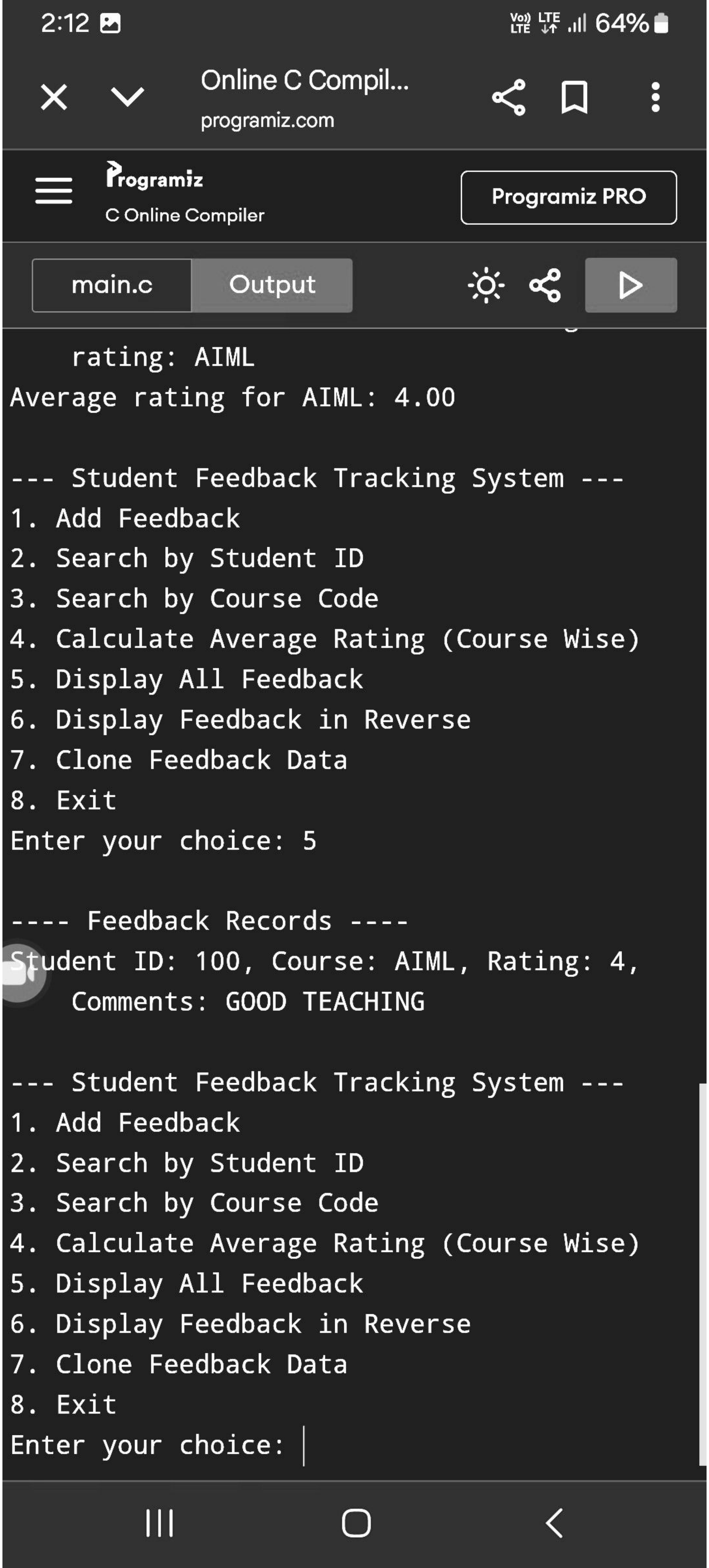
```
--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 1
Enter Student ID: 100
Enter Course Code: AIML
Enter Rating (1-5): 4
Enter Comments: GOOD TEACHING
Feedback added successfully!

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice:
```



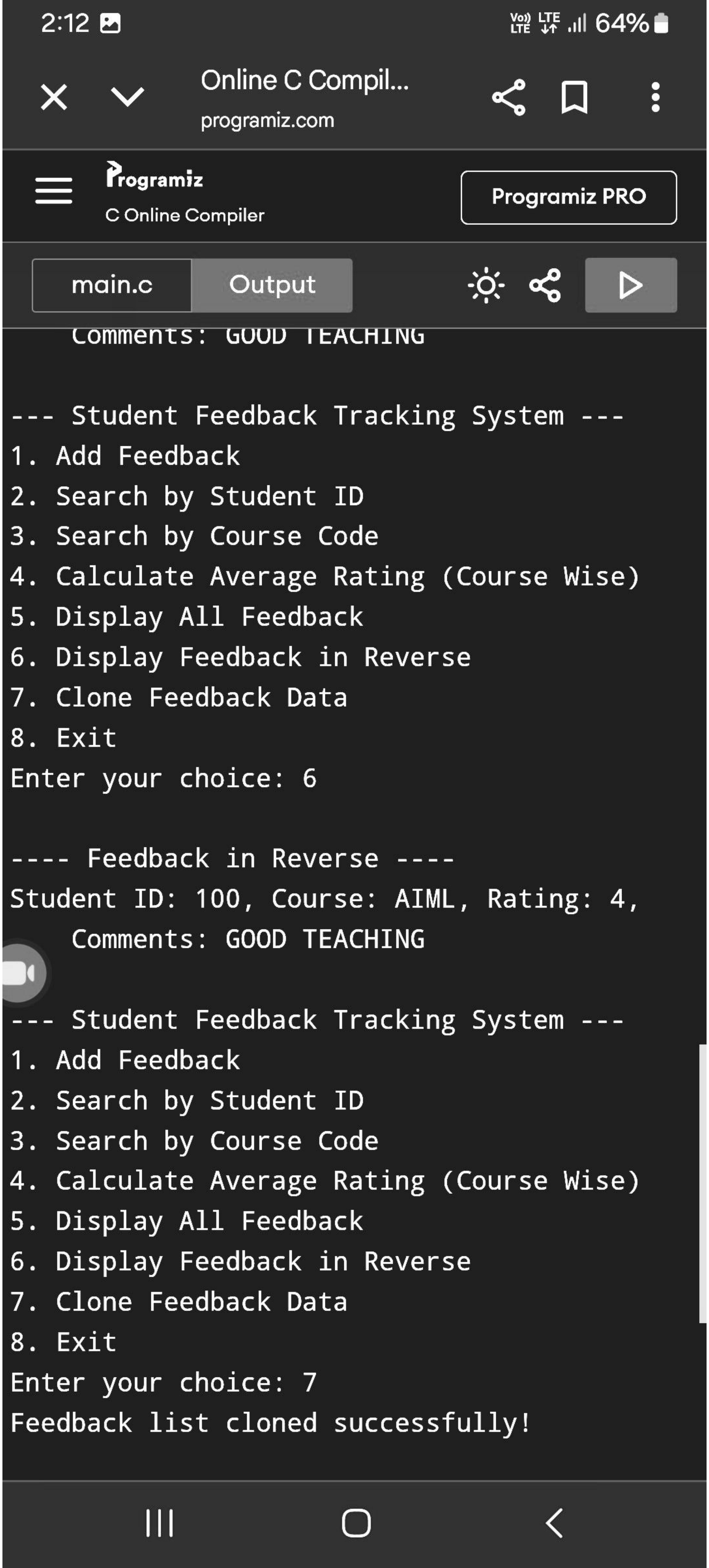
```
--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 2
Enter Student ID to search: 100
Found -> Student ID: 100, Course: AIML,
    Rating: 4, Comments: GOOD TEACHING

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 3
Enter Course Code to search: AIML
Found -> Student ID: 100, Course: AIML,
    Rating: 4, Comments: GOOD TEACHING
```



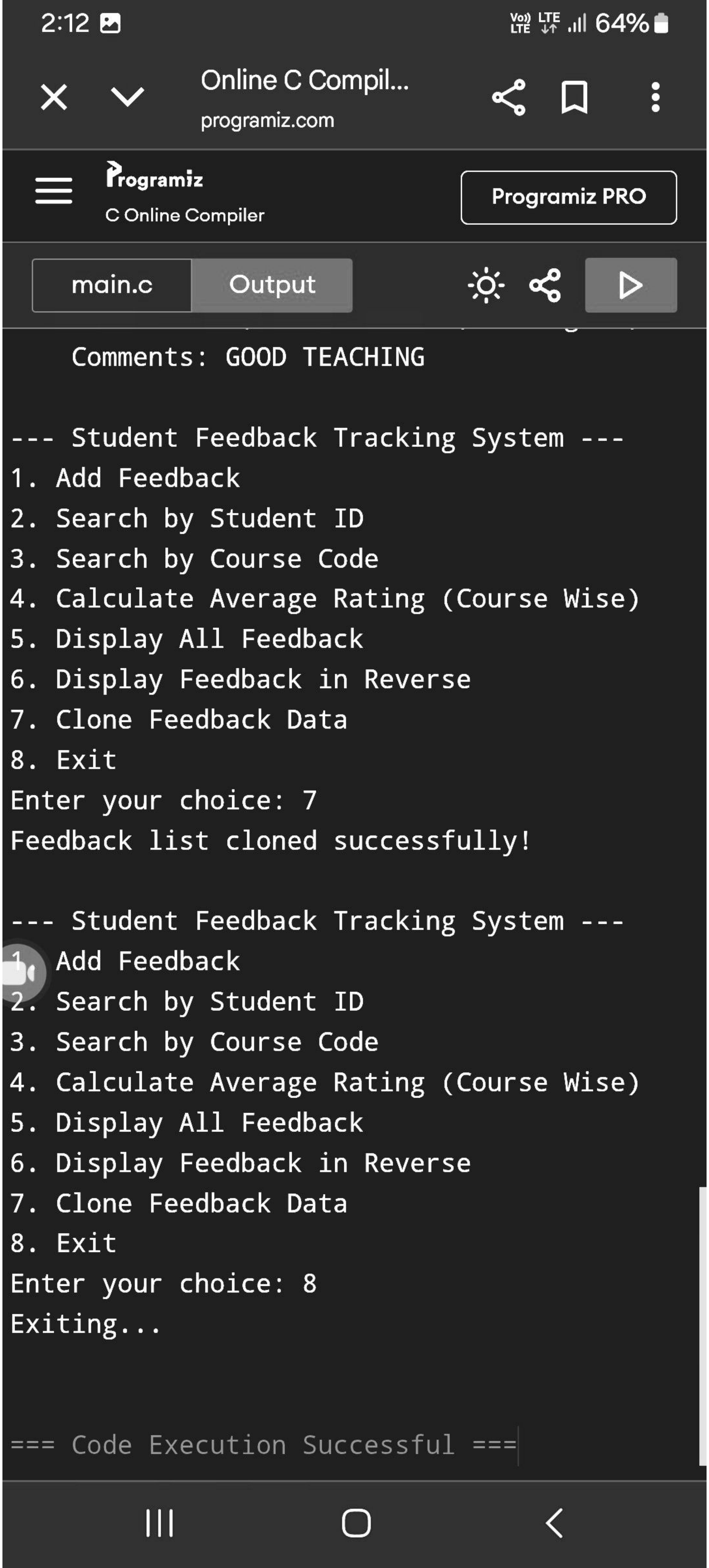
```
    Rating: 4, Comments: GOOD TEACHING

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 4
Enter Course Code to calculate average
    rating: AIML
Average rating for AIML: 4.00

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 5

---- Feedback Records ----
Student ID: 100, Course: AIML, Rating: 4
```



```
    rating: AIML
Average rating for AIML: 4.00

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 5

---- Feedback Records ----
Student ID: 100, Course: AIML, Rating: 4,
    Comments: GOOD TEACHING

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice:
```



```
    Comments: GOOD TEACHING

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 6

---- Feedback in Reverse ----
Student ID: 100, Course: AIML, Rating: 4,
    Comments: GOOD TEACHING

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 7
Feedback list cloned successfully!
```



```
    Comments: GOOD TEACHING

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 7
Feedback list cloned successfully!

--- Student Feedback Tracking System ---
1. Add Feedback
2. Search by Student ID
3. Search by Course Code
4. Calculate Average Rating (Course Wise)
5. Display All Feedback
6. Display Feedback in Reverse
7. Clone Feedback Data
8. Exit
Enter your choice: 8
Exiting...


=== Code Execution Successful ===
```