C

ALGORITHM AND PROGRAMMING

Mid-Terms project
TheTaskTracker

GROUP 6

- MIRZA ANANTA HERMAWAN 2206822212
- DARREL KHAYRU RISYAD 2206820781

BACKGROUND

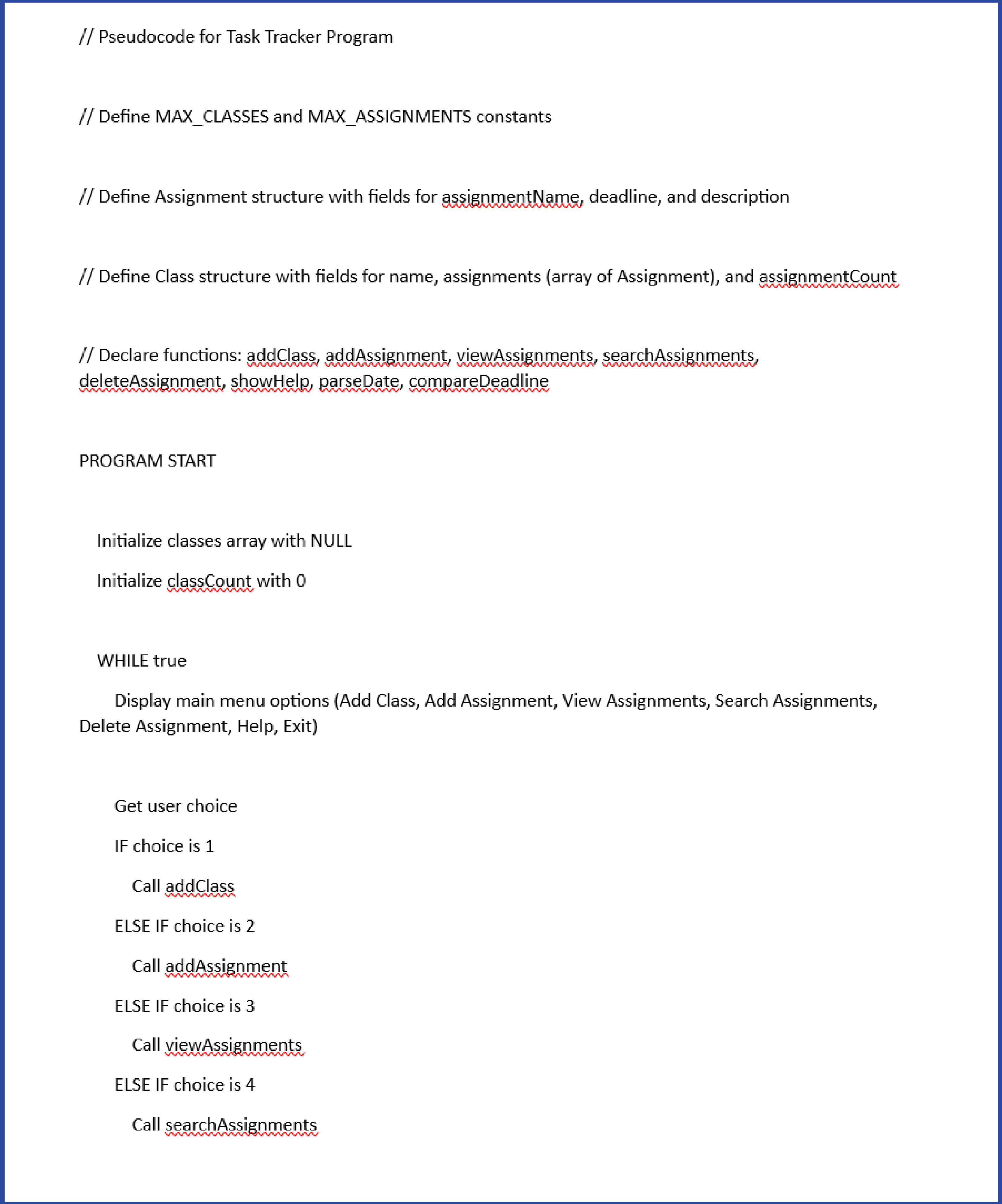
- University students often have multiple subjects to manage each week, with each subject potentially having new assignments or announcements.
- Remembering these details can be challenging, and written notes can get lost or misplaced.
- To address this issue, we developed a code-based solution.
- This program allows students to input their subjects, along with related assignments and announcements.
- Information is organized and can be easily sorted, simplifying academic task management.

ABOUT THE CODE

- Our program focuses on **organizing academic tasks** like assignmentsand deadlines through efficient sorting.
- The system **sorts tasks by deadline or class**, providing a clear view of upcoming deadlines..
- **Completed assignments can be easily removed** from the list, keeping the overview up-to-date.

1

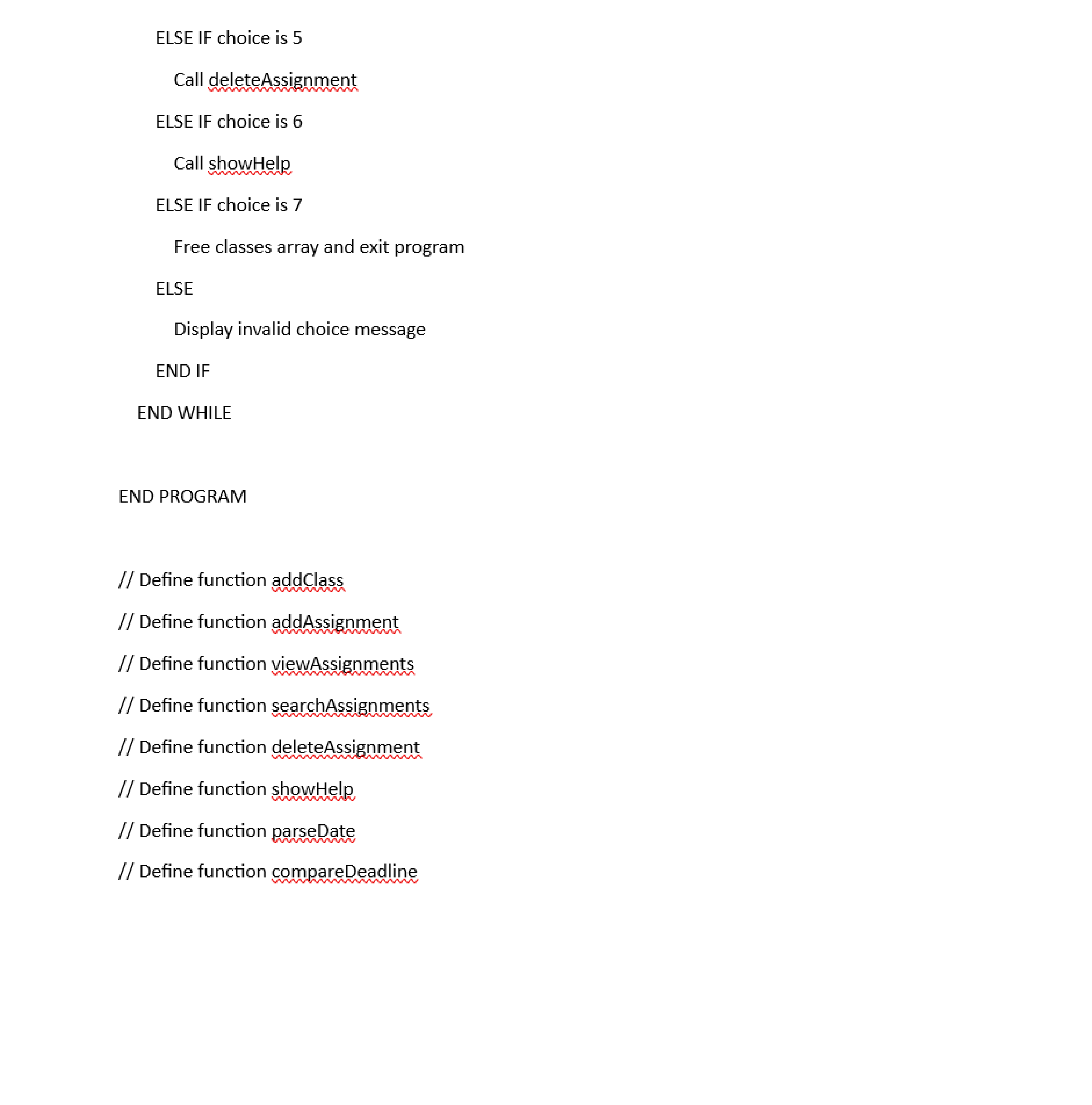
ALGORITHM PSEUDOCODE



```
// Pseudocode for Task Tracker Program

// Define MAX_CLASSES and MAX_ASSIGNMENTS constants

// Define Assignment structure with fields for assignmentName, deadline, and description

// Define Class structure with fields for name, assignments (array of Assignment), and assignmentCount

// Declare functions: addClass, addAssignment, viewAssignments, searchAssignments, deleteAssignment, showHelp, parseDate, compareDeadline

PROGRAM START

  Initialize classes array with NULL
  Initialize classCount with 0

  WHILE true
    Display main menu options (Add Class, Add Assignment, View Assignments, Search Assignments, Delete Assignment, Help, Exit)

    Get user choice
    IF choice is 1
      Call addClass
    ELSE IF choice is 2
      Call addAssignment
    ELSE IF choice is 3
      Call viewAssignments
    ELSE IF choice is 4
      Call searchAssignments
```

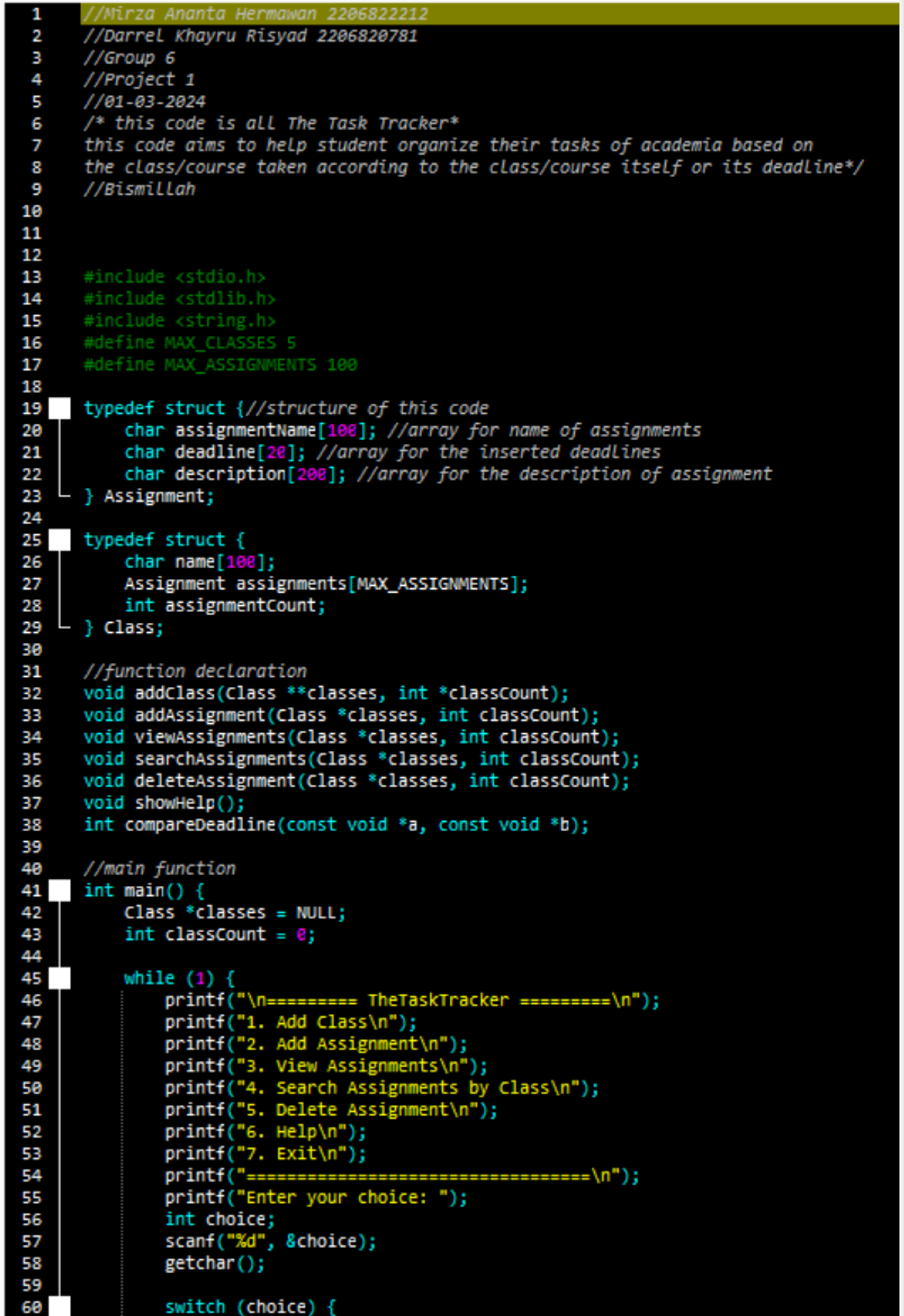
2

ALGORITHM PSEUDOCODE

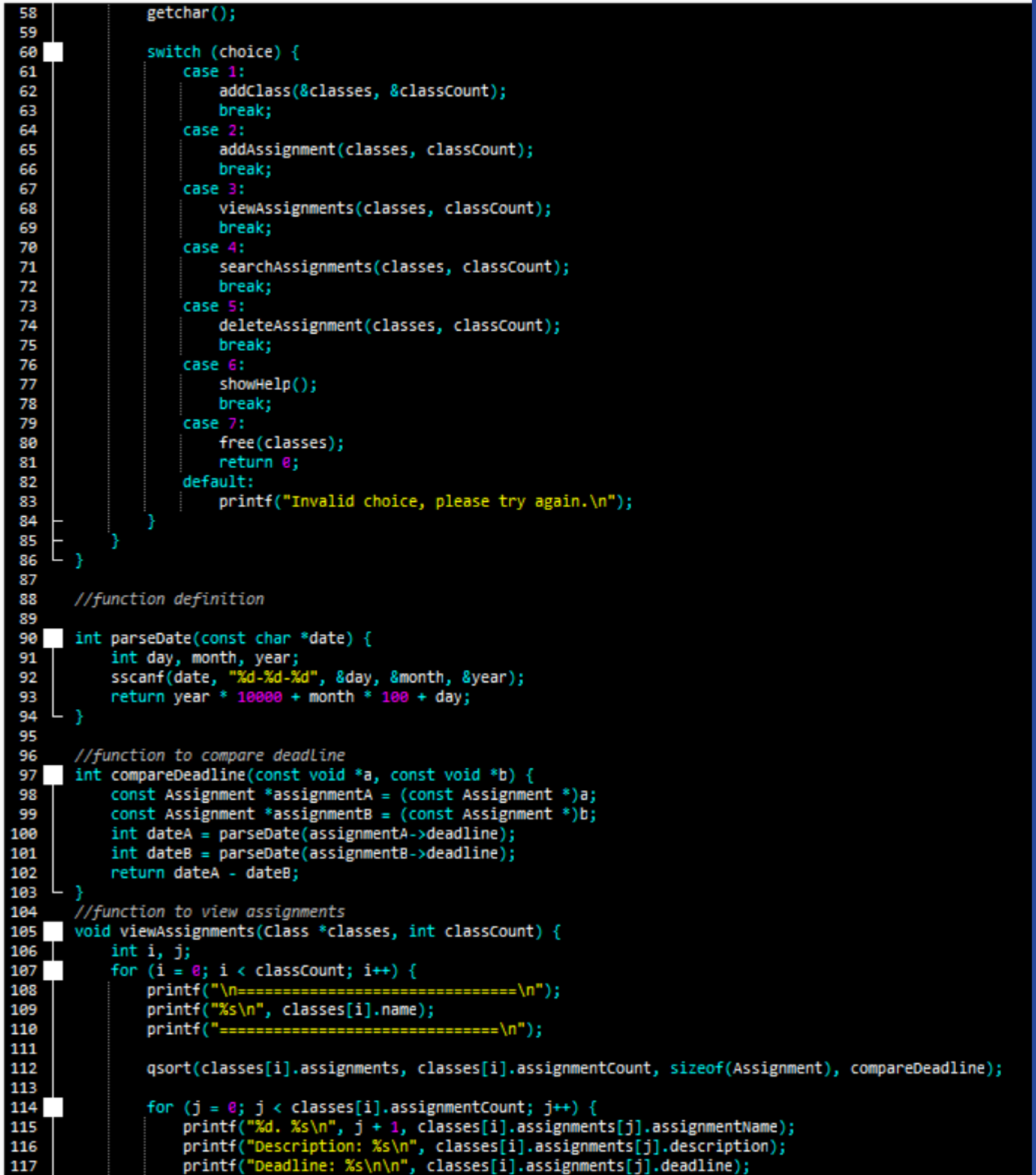


```
    ELSE IF choice is 5
      Call deleteAssignment
    ELSE IF choice is 6
      Call showHelp
    ELSE IF choice is 7
      Free classes array and exit program
    ELSE
      Display invalid choice message
    END IF
  END WHILE

END PROGRAM

// Define function addClass
// Define function addAssignment
// Define function viewAssignments
// Define function searchAssignments
// Define function deleteAssignment
// Define function showHelp
// Define function parseDate
// Define function compareDeadline
```

CODE



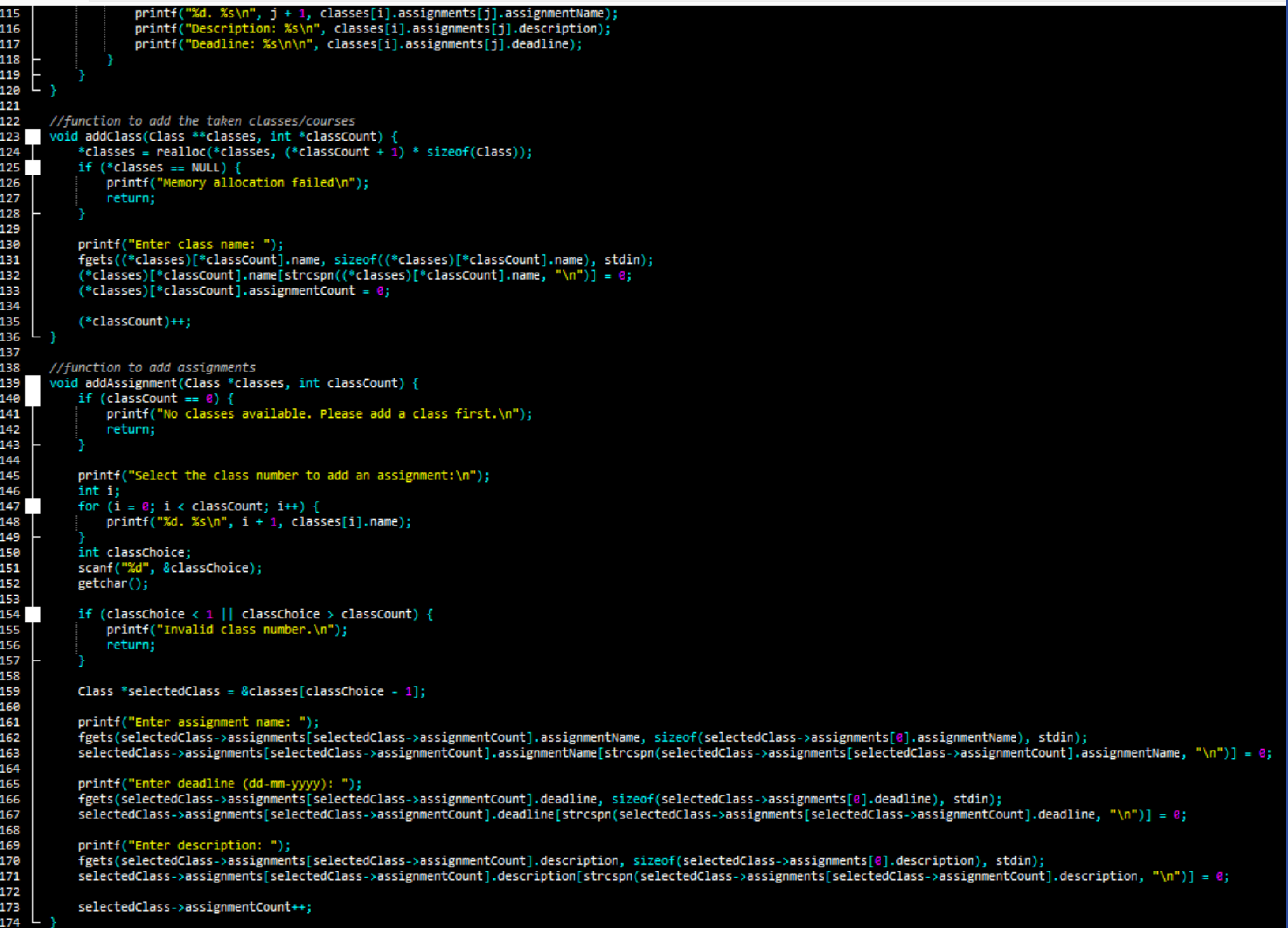
```c
//Mirza Ananta Hermawan 2206822212
//Darrel Khayru Risyad 2206820781
//Group 6
//Project 1
//01-03-2024
/* this code is all The Task Tracker*
this code aims to help student organize their tasks of academia based on
the class/course taken according to the class/course itself or its deadline*/
//Bismillah



#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_CLASSES 5
#define MAX_ASSIGNMENTS 100

typedef struct {//structure of this code
    char assignmentName[100]; //array for name of assignments
    char deadline[20]; //array for the inserted deadlines
    char description[200]; //array for the description of assignment
} Assignment;

typedef struct {
    char name[100];
    Assignment assignments[MAX_ASSIGNMENTS];
    int assignmentCount;
} Class;

//function declaration
void addClass(Class **classes, int *classCount);
void addAssignment(Class *classes, int classCount);
void viewAssignments(Class *classes, int classCount);
void searchAssignments(Class *classes, int classCount);
void deleteAssignment(Class *classes, int classCount);
void showHelp();
int compareDeadline(const void *a, const void *b);

//main function
int main() {
    Class *classes = NULL;
    int classCount = 0;

    while (1) {
        printf("\n========= TheTaskTracker =========\n");
        printf("1. Add Class\n");
        printf("2. Add Assignment\n");
        printf("3. View Assignments\n");
        printf("4. Search Assignments by Class\n");
        printf("5. Delete Assignment\n");
        printf("6. Help\n");
        printf("7. Exit\n");
        printf("==================================\n");
        printf("Enter your choice: ");
        int choice;
        scanf("%d", &choice);
        getchar();

        switch (choice) {
```



```c
        getchar();

        switch (choice) {
            case 1:
                addClass(&classes, &classCount);
                break;
            case 2:
                addAssignment(classes, classCount);
                break;
            case 3:
                viewAssignments(classes, classCount);
                break;
            case 4:
                searchAssignments(classes, classCount);
                break;
            case 5:
                deleteAssignment(classes, classCount);
                break;
            case 6:
                showHelp();
                break;
            case 7:
                free(classes);
                return 0;
            default:
                printf("Invalid choice, please try again.\n");
        }
    }
}

//function definition

int parseDate(const char *date) {
    int day, month, year;
    sscanf(date, "%d-%d-%d", &day, &month, &year);
    return year * 10000 + month * 100 + day;
}

//function to compare deadline
int compareDeadline(const void *a, const void *b) {
    const Assignment *assignmentA = (const Assignment *)a;
    const Assignment *assignmentB = (const Assignment *)b;
    int dateA = parseDate(assignmentA->deadline);
    int dateB = parseDate(assignmentB->deadline);
    return dateA - dateB;
}
//function to view assignments
void viewAssignments(Class *classes, int classCount) {
    int i, j;
    for (i = 0; i < classCount; i++) {
        printf("\n===============================\n");
        printf("%s\n", classes[i].name);
        printf("===============================\n");

        qsort(classes[i].assignments, classes[i].assignmentCount, sizeof(Assignment), compareDeadline);

        for (j = 0; j < classes[i].assignmentCount; j++) {
            printf("%d. %s\n", j + 1, classes[i].assignments[j].assignmentName);
            printf("Description: %s\n", classes[i].assignments[j].description);
            printf("Deadline: %s\n\n", classes[i].assignments[j].deadline);
```

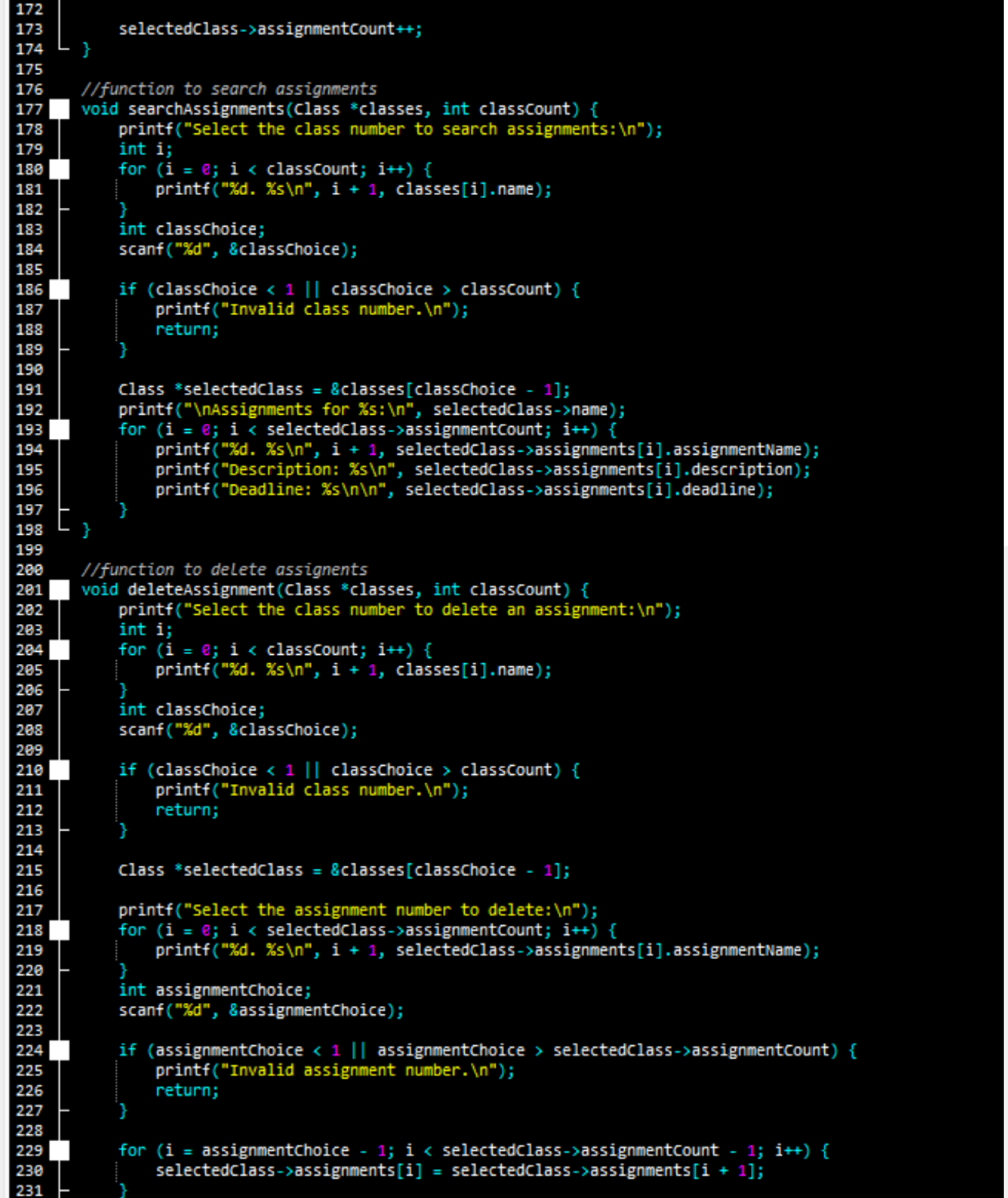
3

THE CODE

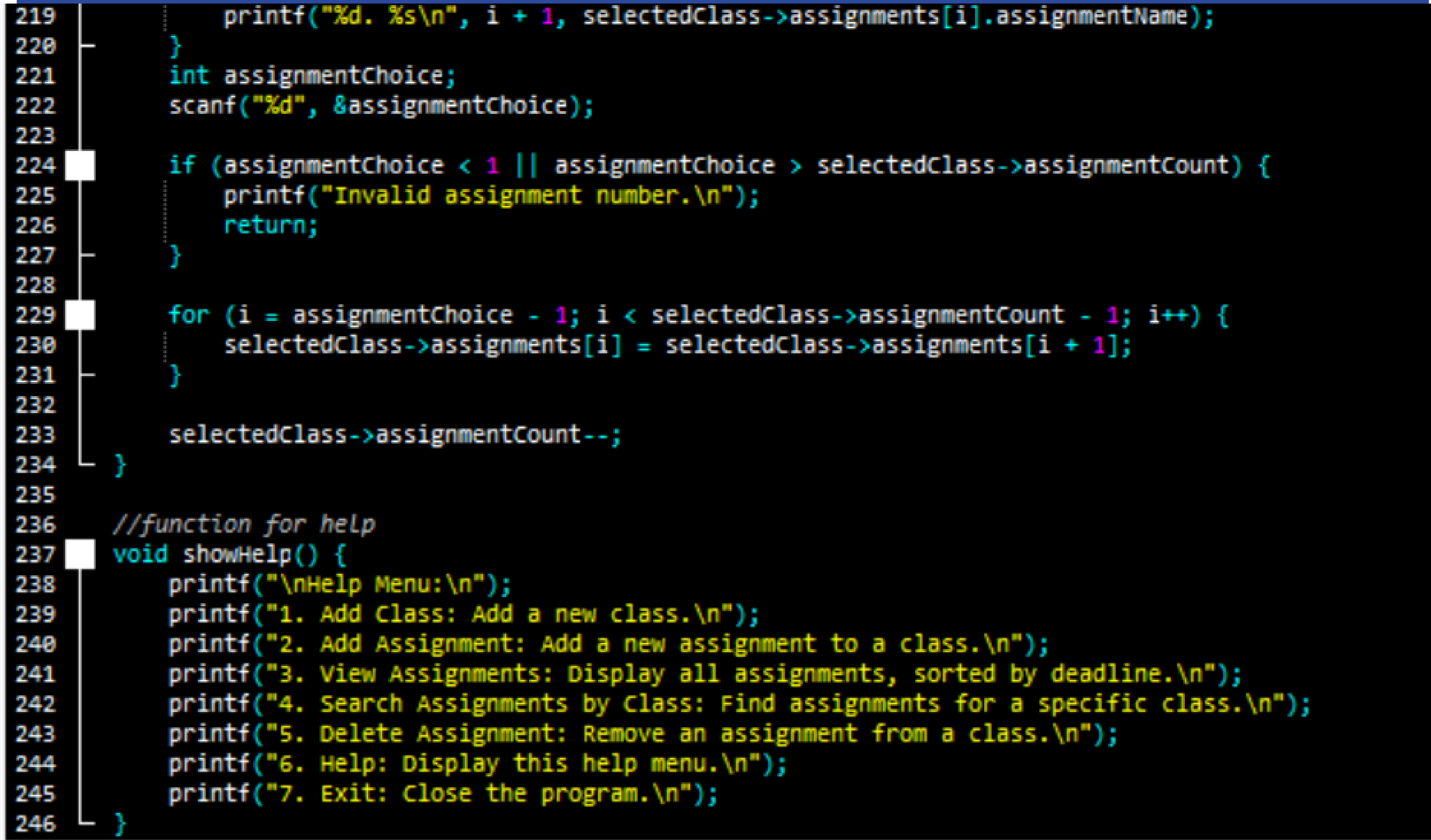


```c
            printf("%d. %s\n", j + 1, classes[i].assignments[j].assignmentName);
            printf("Description: %s\n", classes[i].assignments[j].description);
            printf("Deadline: %s\n\n", classes[i].assignments[j].deadline);
        }
    }
}

//function to add the taken classes/courses
void addClass(Class **classes, int *classCount) {
    *classes = realloc(*classes, (*classCount + 1) * sizeof(Class));
    if (*classes == NULL) {
        printf("Memory allocation failed\n");
        return;
    }

    printf("Enter class name: ");
    fgets((*classes)[*classCount].name, sizeof((*classes)[*classCount].name), stdin);
    (*classes)[*classCount].name[strcspn((*classes)[*classCount].name, "\n")] = 0;
    (*classes)[*classCount].assignmentCount = 0;

    (*classCount)++;
}

//function to add assignments
void addAssignment(Class *classes, int classCount) {
    if (classCount == 0) {
        printf("No classes available. Please add a class first.\n");
        return;
    }

    printf("Select the class number to add an assignment:\n");
    int i;
    for (i = 0; i < classCount; i++) {
        printf("%d. %s\n", i + 1, classes[i].name);
    }
    int classChoice;
    scanf("%d", &classChoice);
    getchar();

    if (classChoice < 1 || classChoice > classCount) {
        printf("Invalid class number.\n");
        return;
    }

    Class *selectedClass = &classes[classChoice - 1];

    printf("Enter assignment name: ");
    fgets(selectedClass->assignments[selectedClass->assignmentCount].assignmentName, sizeof(selectedClass->assignments[0].assignmentName), stdin);
    selectedClass->assignments[selectedClass->assignmentCount].assignmentName[strcspn(selectedClass->assignments[selectedClass->assignmentCount].assignmentName, "\n")] = 0;

    printf("Enter deadline (dd-mm-yyyy): ");
    fgets(selectedClass->assignments[selectedClass->assignmentCount].deadline, sizeof(selectedClass->assignments[0].deadline), stdin);
    selectedClass->assignments[selectedClass->assignmentCount].deadline[strcspn(selectedClass->assignments[selectedClass->assignmentCount].deadline, "\n")] = 0;

    printf("Enter description: ");
    fgets(selectedClass->assignments[selectedClass->assignmentCount].description, sizeof(selectedClass->assignments[0].description), stdin);
    selectedClass->assignments[selectedClass->assignmentCount].description[strcspn(selectedClass->assignments[selectedClass->assignmentCount].description, "\n")] = 0;

    selectedClass->assignmentCount++;
}
```

THE CODE



```c

    selectedClass->assignmentCount++;
}

//function to search assignments
void searchAssignments(Class *classes, int classCount) {
    printf("Select the class number to search assignments:\n");
    int i;
    for (i = 0; i < classCount; i++) {
        printf("%d. %s\n", i + 1, classes[i].name);
    }
    int classChoice;
    scanf("%d", &classChoice);

    if (classChoice < 1 || classChoice > classCount) {
        printf("Invalid class number.\n");
        return;
    }

    Class *selectedClass = &classes[classChoice - 1];
    printf("\nAssignments for %s:\n", selectedClass->name);
    for (i = 0; i < selectedClass->assignmentCount; i++) {
        printf("%d. %s\n", i + 1, selectedClass->assignments[i].assignmentName);
        printf("Description: %s\n", selectedClass->assignments[i].description);
        printf("Deadline: %s\n\n", selectedClass->assignments[i].deadline);
    }
}

//function to delete assignents
void deleteAssignment(Class *classes, int classCount) {
    printf("Select the class number to delete an assignment:\n");
    int i;
    for (i = 0; i < classCount; i++) {
        printf("%d. %s\n", i + 1, classes[i].name);
    }
    int classChoice;
    scanf("%d", &classChoice);

    if (classChoice < 1 || classChoice > classCount) {
        printf("Invalid class number.\n");
        return;
    }

    Class *selectedClass = &classes[classChoice - 1];

    printf("Select the assignment number to delete:\n");
    for (i = 0; i < selectedClass->assignmentCount; i++) {
        printf("%d. %s\n", i + 1, selectedClass->assignments[i].assignmentName);
    }
    int assignmentChoice;
    scanf("%d", &assignmentChoice);

    if (assignmentChoice < 1 || assignmentChoice > selectedClass->assignmentCount) {
        printf("Invalid assignment number.\n");
        return;
    }

    for (i = assignmentChoice - 1; i < selectedClass->assignmentCount - 1; i++) {
        selectedClass->assignments[i] = selectedClass->assignments[i + 1];
    }
```

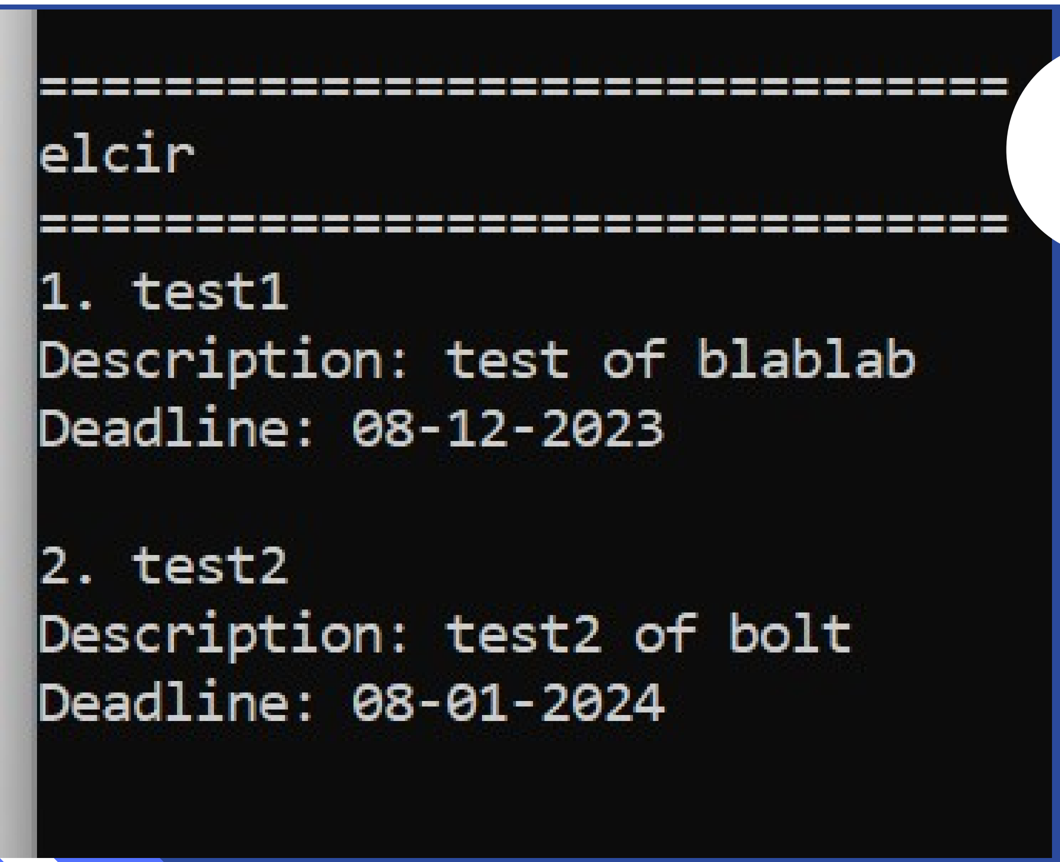


```c
        printf("%d. %s\n", i + 1, selectedClass->assignments[i].assignmentName);
    }
    int assignmentChoice;
    scanf("%d", &assignmentChoice);

    if (assignmentChoice < 1 || assignmentChoice > selectedClass->assignmentCount) {
        printf("Invalid assignment number.\n");
        return;
    }

    for (i = assignmentChoice - 1; i < selectedClass->assignmentCount - 1; i++) {
        selectedClass->assignments[i] = selectedClass->assignments[i + 1];
    }
```

```c

    selectedClass->assignmentCount--;
}

//function for help
void showHelp() {
    printf("\nHelp Menu:\n");
    printf("1. Add Class: Add a new class.\n");
    printf("2. Add Assignment: Add a new assignment to a class.\n");
    printf("3. View Assignments: Display all assignments, sorted by deadline.\n");
    printf("4. Search Assignments by Class: Find assignments for a specific class.\n");
    printf("5. Delete Assignment: Remove an assignment from a class.\n");
    printf("6. Help: Display this help menu.\n");
    printf("7. Exit: Close the program.\n");
}
```

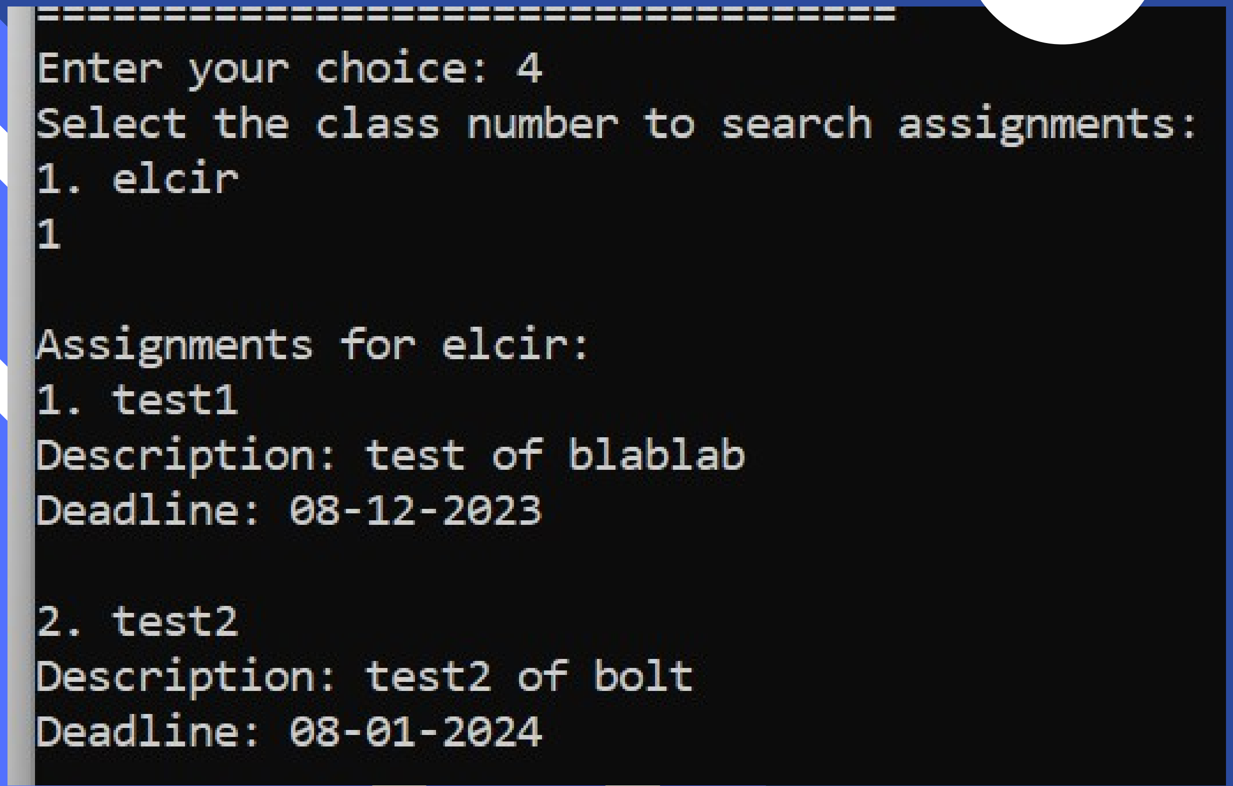
PROGRAM IN ACTION

1



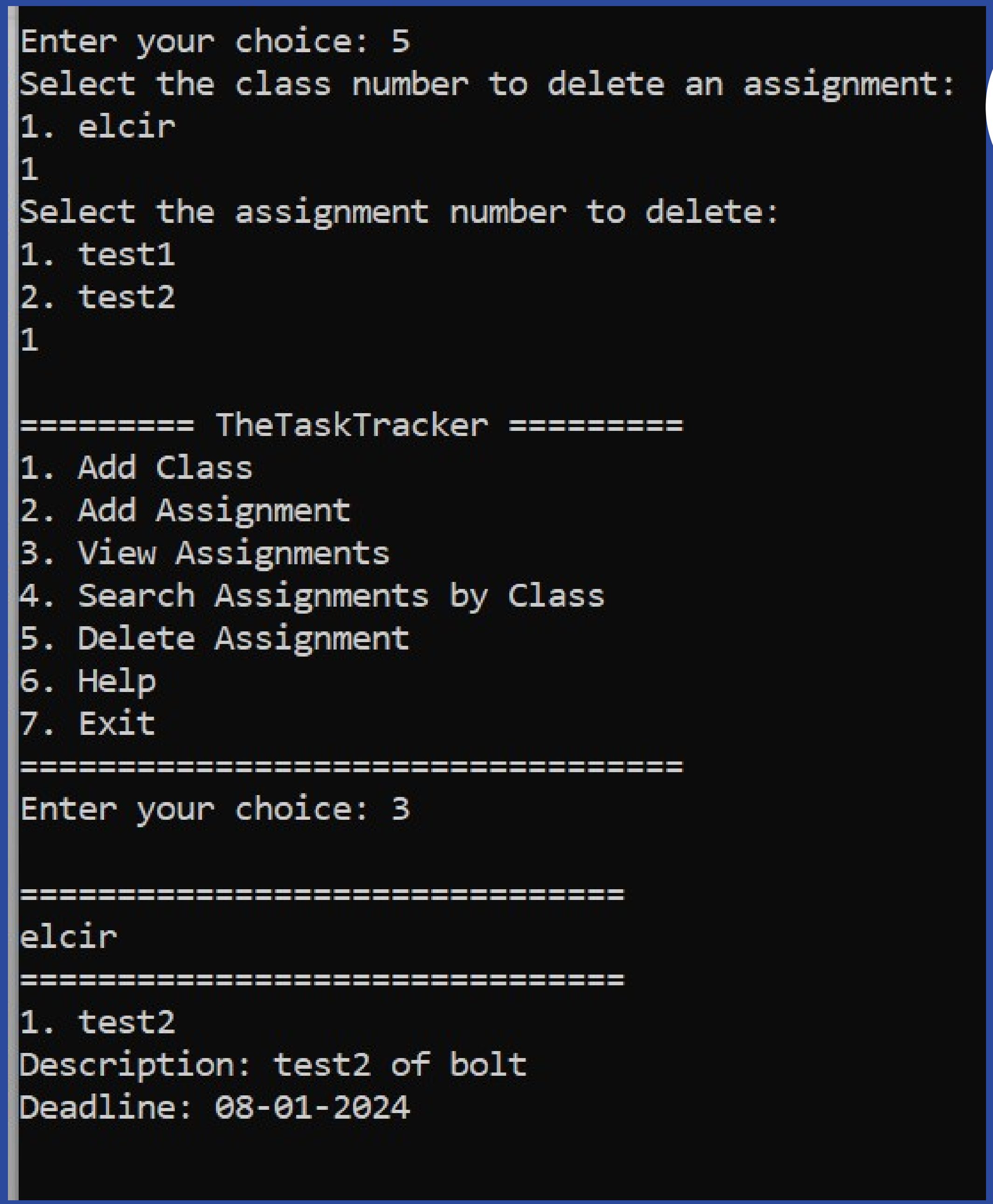
```
===============================
elcir
===============================
1. test1
Description: test of blablab
Deadline: 08-12-2023

2. test2
Description: test2 of bolt
Deadline: 08-01-2024
```

2



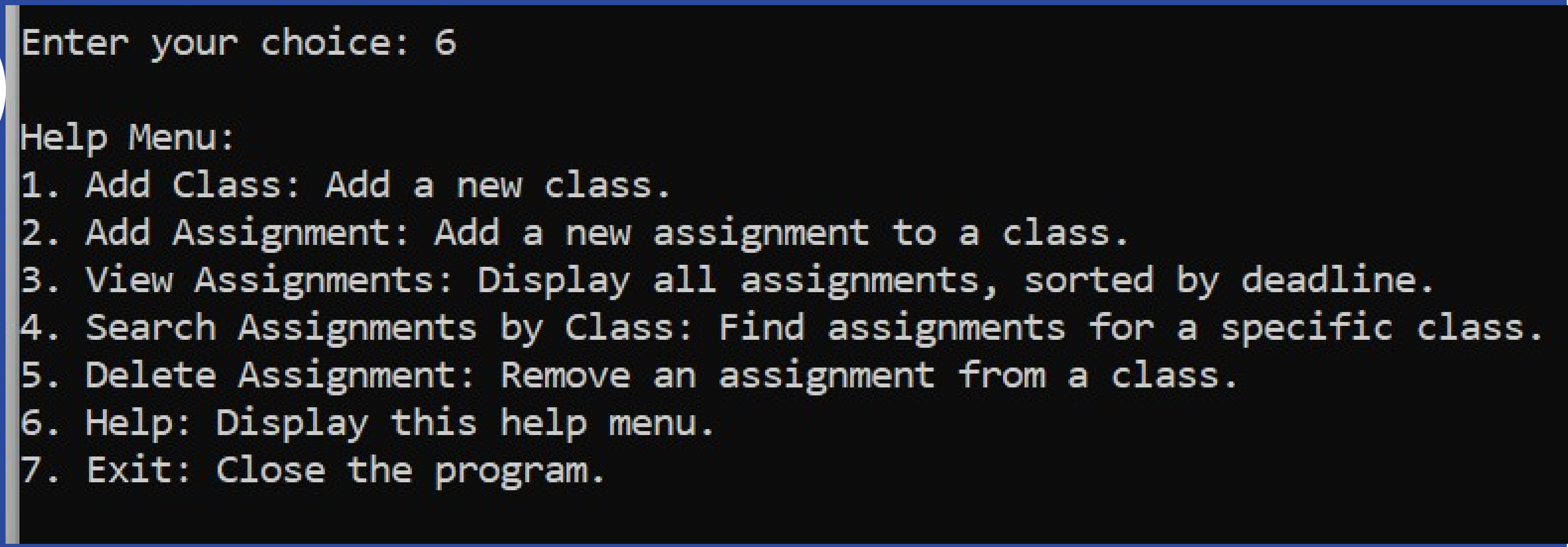
```
==================================
Enter your choice: 4
Select the class number to search assignments:
1. elcir
1

Assignments for elcir:
1. test1
Description: test of blablab
Deadline: 08-12-2023

2. test2
Description: test2 of bolt
Deadline: 08-01-2024
```

3



```
Enter your choice: 5
Select the class number to delete an assignment:
1. elcir
1
Select the assignment number to delete:
1. test1
2. test2
1

========= TheTaskTracker =========
1. Add Class
2. Add Assignment
3. View Assignments
4. Search Assignments by Class
5. Delete Assignment
6. Help
7. Exit
==================================
Enter your choice: 3

===============================
elcir
===============================
1. test2
Description: test2 of bolt
Deadline: 08-01-2024
```

4



```
Enter your choice: 6

Help Menu:
1. Add Class: Add a new class.
2. Add Assignment: Add a new assignment to a class.
3. View Assignments: Display all assignments, sorted by deadline.
4. Search Assignments by Class: Find assignments for a specific class.
5. Delete Assignment: Remove an assignment from a class.
6. Help: Display this help menu.
7. Exit: Close the program.
```

THANK YOU

TheTaskTracker

GROUP 6

- MIRZA ANANTA HERMAWAN 2206822212
- DARREL KHAYRU RISYAD 2206820781