

ALGORITHM AND PROGRAMMING

Mid-Terms project
TheTaskTracker

GROUP 6

- MIRZA ANANTA HERMAWAN 2206822212
 - DARREL KHAYRU RISYAD 2206820781
- 

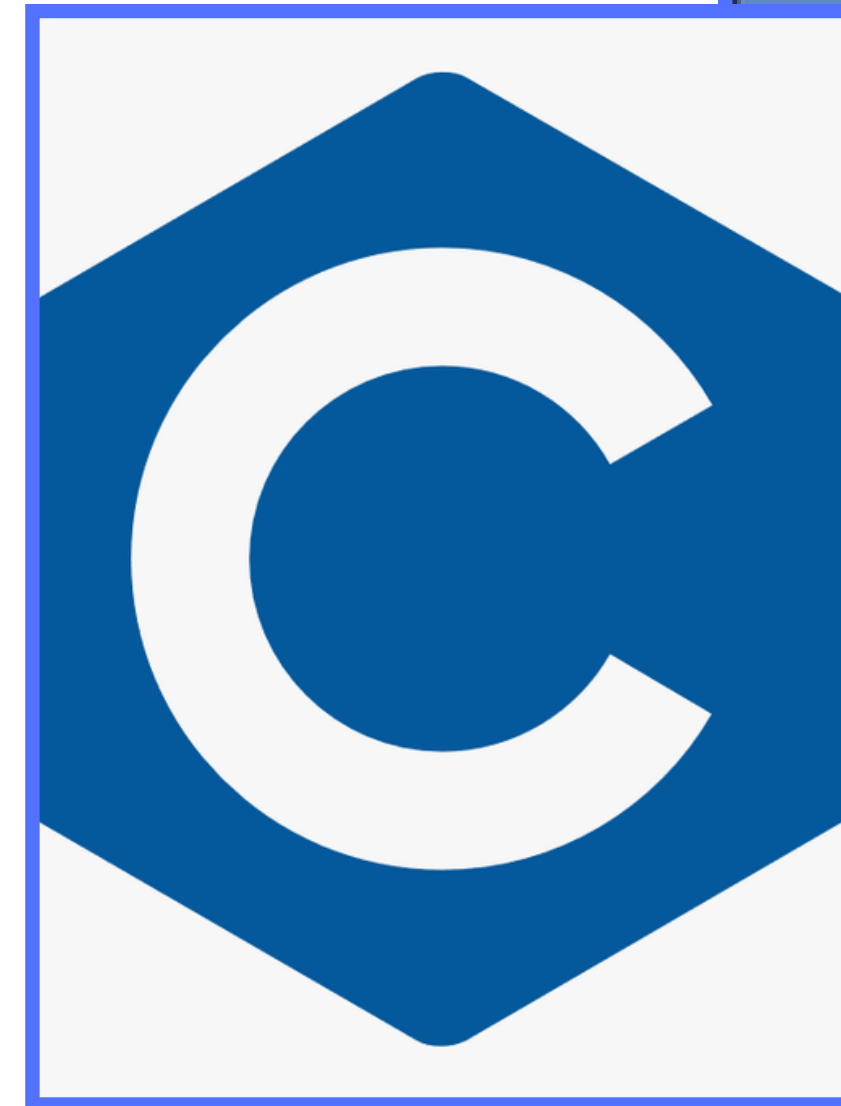


BACKGROUND

- 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 assignments and 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
```

C O D E

```
1 //Mirza Ananta Hermawan 2206822212
2 //Darrel Khayru Risyad 2206820781
3 //Group 6
4 //Project 1
5 //01-03-2024
6 /* this code is all The Task Tracker*
7 this code aims to help student organize their tasks of academia based on
8 the class/course taken according to the class/course itself or its deadline*/
9 //Bismillah
10
11
12
13 #include <stdio.h>
14 #include <stdlib.h>
15 #include <string.h>
16 #define MAX_CLASSES 5
17 #define MAX_ASSIGNMENTS 100
18
19 typedef struct { //structure of this code
20     char assignmentName[100]; //array for name of assignments
21     char deadline[20]; //array for the inserted deadlines
22     char description[200]; //array for the description of assignment
23 } Assignment;
24
25 typedef struct {
26     char name[100];
27     Assignment assignments[MAX_ASSIGNMENTS];
28     int assignmentCount;
29 } Class;
30
31 //function declaration
32 void addClass(Class **classes, int *classCount);
33 void addAssignment(Class *classes, int classCount);
34 void viewAssignments(Class *classes, int classCount);
35 void searchAssignments(Class *classes, int classCount);
36 void deleteAssignment(Class *classes, int classCount);
37 void showHelp();
38 int compareDeadline(const void *a, const void *b);
39
40 //main function
41 int main() {
42     Class *classes = NULL;
43     int classCount = 0;
44
45     while (1) {
46         printf("\n===== TheTaskTracker =====\n");
47         printf("1. Add Class\n");
48         printf("2. Add Assignment\n");
49         printf("3. View Assignments\n");
50         printf("4. Search Assignments by Class\n");
51         printf("5. Delete Assignment\n");
52         printf("6. Help\n");
53         printf("7. Exit\n");
54         printf("=====\n");
55         printf("Enter your choice: ");
56         int choice;
57         scanf("%d", &choice);
58         getchar();
59
60         switch (choice) {
```

```
58         getchar();
59
60         switch (choice) {
61             case 1:
62                 addClass(&classes, &classCount);
63                 break;
64             case 2:
65                 addAssignment(classes, classCount);
66                 break;
67             case 3:
68                 viewAssignments(classes, classCount);
69                 break;
70             case 4:
71                 searchAssignments(classes, classCount);
72                 break;
73             case 5:
74                 deleteAssignment(classes, classCount);
75                 break;
76             case 6:
77                 showHelp();
78                 break;
79             case 7:
80                 free(classes);
81                 return 0;
82             default:
83                 printf("Invalid choice, please try again.\n");
84         }
85     }
86 }
87
88 //function definition
89
90 int parseDate(const char *date) {
91     int day, month, year;
92     sscanf(date, "%d-%d-%d", &day, &month, &year);
93     return year * 10000 + month * 100 + day;
94 }
95
96 //function to compare deadline
97 int compareDeadline(const void *a, const void *b) {
98     const Assignment *assignmentA = (const Assignment *)a;
99     const Assignment *assignmentB = (const Assignment *)b;
100     int dateA = parseDate(assignmentA->deadline);
101     int dateB = parseDate(assignmentB->deadline);
102     return dateA - dateB;
103 }
104
105 //function to view assignments
106 void viewAssignments(Class *classes, int classCount) {
107     int i, j;
108     for (i = 0; i < classCount; i++) {
109         printf("\n=====\n");
110         printf("%s\n", classes[i].name);
111         printf("=====\n");
112
113         qsort(classes[i].assignments, classes[i].assignmentCount, sizeof(Assignment), compareDeadline);
114
115         for (j = 0; j < classes[i].assignmentCount; j++) {
116             printf("%d. %s\n", j + 1, classes[i].assignments[j].assignmentName);
117             printf("Description: %s\n", classes[i].assignments[j].description);
118             printf("Deadline: %s\n", classes[i].assignments[j].deadline);
119         }
120     }
121 }
```


3

THE CODE

```
115         printf("%d. %s\n", j + 1, classes[i].assignments[j].assignmentName);
116         printf("Description: %s\n", classes[i].assignments[j].description);
117         printf("Deadline: %s\n\n", classes[i].assignments[j].deadline);
118     }
119 }
120 }
121
122 //function to add the taken classes/courses
123 void addClass(Class **classes, int *classCount) {
124     *classes = realloc(*classes, (*classCount + 1) * sizeof(Class));
125     if (*classes == NULL) {
126         printf("Memory allocation failed\n");
127         return;
128     }
129
130     printf("Enter class name: ");
131     fgets((*classes)[*classCount].name, sizeof((*classes)[*classCount].name), stdin);
132     (*classes)[*classCount].name[strcspn((*classes)[*classCount].name, "\n")] = 0;
133     (*classes)[*classCount].assignmentCount = 0;
134
135     (*classCount)++;
136 }
137
138 //function to add assignments
139 void addAssignment(Class *classes, int classCount) {
140     if (classCount == 0) {
141         printf("No classes available. Please add a class first.\n");
142         return;
143     }
144
145     printf("Select the class number to add an assignment:\n");
146     int i;
147     for (i = 0; i < classCount; i++) {
148         printf("%d. %s\n", i + 1, classes[i].name);
149     }
150     int classChoice;
151     scanf("%d", &classChoice);
152     getchar();
153
154     if (classChoice < 1 || classChoice > classCount) {
155         printf("Invalid class number.\n");
156         return;
157     }
158
159     Class *selectedClass = &classes[classChoice - 1];
160
161     printf("Enter assignment name: ");
162     fgets(selectedClass->assignments[selectedClass->assignmentCount].assignmentName, sizeof(selectedClass->assignments[0].assignmentName), stdin);
163     selectedClass->assignments[selectedClass->assignmentCount].assignmentName[strcspn(selectedClass->assignments[selectedClass->assignmentCount].assignmentName, "\n")] = 0;
164
165     printf("Enter deadline (dd-mm-yyyy): ");
166     fgets(selectedClass->assignments[selectedClass->assignmentCount].deadline, sizeof(selectedClass->assignments[0].deadline), stdin);
167     selectedClass->assignments[selectedClass->assignmentCount].deadline[strcspn(selectedClass->assignments[selectedClass->assignmentCount].deadline, "\n")] = 0;
168
169     printf("Enter description: ");
170     fgets(selectedClass->assignments[selectedClass->assignmentCount].description, sizeof(selectedClass->assignments[0].description), stdin);
171     selectedClass->assignments[selectedClass->assignmentCount].description[strcspn(selectedClass->assignments[selectedClass->assignmentCount].description, "\n")] = 0;
172
173     selectedClass->assignmentCount++;
174 }
```

THE CODE

```
172     selectedClass->assignmentCount++;
173 }
174
175 //function to search assignments
176 void searchAssignments(Class *classes, int classCount) {
177     printf("Select the class number to search assignments:\n");
178     int i;
179     for (i = 0; i < classCount; i++) {
180         printf("%d. %s\n", i + 1, classes[i].name);
181     }
182     int classChoice;
183     scanf("%d", &classChoice);
184
185     if (classChoice < 1 || classChoice > classCount) {
186         printf("Invalid class number.\n");
187         return;
188     }
189
190     Class *selectedClass = &classes[classChoice - 1];
191     printf("\nAssignments for %s:\n", selectedClass->name);
192     for (i = 0; i < selectedClass->assignmentCount; i++) {
193         printf("%d. %s\n", i + 1, selectedClass->assignments[i].assignmentName);
194         printf("Description: %s\n", selectedClass->assignments[i].description);
195         printf("Deadline: %s\n\n", selectedClass->assignments[i].deadline);
196     }
197 }
198
199 //function to delete assignments
200 void deleteAssignment(Class *classes, int classCount) {
201     printf("Select the class number to delete an assignment:\n");
202     int i;
203     for (i = 0; i < classCount; i++) {
204         printf("%d. %s\n", i + 1, classes[i].name);
205     }
206     int classChoice;
207     scanf("%d", &classChoice);
208
209     if (classChoice < 1 || classChoice > classCount) {
210         printf("Invalid class number.\n");
211         return;
212     }
213
214     Class *selectedClass = &classes[classChoice - 1];
215
216     printf("Select the assignment number to delete:\n");
217     for (i = 0; i < selectedClass->assignmentCount; i++) {
218         printf("%d. %s\n", i + 1, selectedClass->assignments[i].assignmentName);
219     }
220     int assignmentChoice;
221     scanf("%d", &assignmentChoice);
222
223     if (assignmentChoice < 1 || assignmentChoice > selectedClass->assignmentCount) {
224         printf("Invalid assignment number.\n");
225         return;
226     }
227
228     for (i = assignmentChoice - 1; i < selectedClass->assignmentCount - 1; i++) {
229         selectedClass->assignments[i] = selectedClass->assignments[i + 1];
230     }
231 }
```

```
219     printf("%d. %s\n", i + 1, selectedClass->assignments[i].assignmentName);
220 }
221 int assignmentChoice;
222 scanf("%d", &assignmentChoice);
223
224 if (assignmentChoice < 1 || assignmentChoice > selectedClass->assignmentCount) {
225     printf("Invalid assignment number.\n");
226     return;
227 }
228
229 for (i = assignmentChoice - 1; i < selectedClass->assignmentCount - 1; i++) {
230     selectedClass->assignments[i] = selectedClass->assignments[i + 1];
231 }
232
233 selectedClass->assignmentCount--;
234 }
235
236 //function for help
237 void showHelp() {
238     printf("\nHelp Menu:\n");
239     printf("1. Add Class: Add a new class.\n");
240     printf("2. Add Assignment: Add a new assignment to a class.\n");
241     printf("3. View Assignments: Display all assignments, sorted by deadline.\n");
242     printf("4. Search Assignments by Class: Find assignments for a specific class.\n");
243     printf("5. Delete Assignment: Remove an assignment from a class.\n");
244     printf("6. Help: Display this help menu.\n");
245     printf("7. Exit: Close the program.\n");
246 }
```


PROGRAM IN ACTION

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

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

1

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
```

```
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

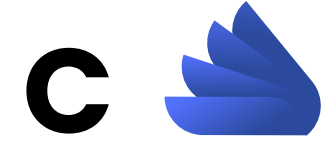
=====
elcir
=====
1. test2
Description: test2 of bolt
Deadline: 08-01-2024
```

3

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.
```

A dark grey line starts from the top left, goes horizontally to the right, then diagonally down to a solid dark grey circle.

THANK YOU

TheTaskTracker

GROUP 6

- MIRZA ANANTA HERMAWAN 2206822212
 - DARREL KHAYRU RISYAD 2206820781
- 
- Large blue geometric shapes on the right side of the slide, including a large triangle pointing left and a smaller triangle pointing right, both with white outlines.