

ASSIGNMENT-4

Purpose:

- This program processes data read from specific files to handle user information. Various tasks are accomplished through functions: finding the first letters of user names and surnames, finding identification numbers of users with names or surnames starting with a specific letter, calculating the average grade of a specific user, etc.

Code Structure and Functions:

1-) first_initial(FILE *id_fp, int id):

- This function returns the first letter of the name of a user with a given identification number from the "second 1.txt" file.
- The file pointer "id_fp" is used to open the file and read the data.
- A loop checks the user identification number until the end of the file.
- Once the identification number of the desired user is found, the first letter of their name is returned.

2-) last_initial(FILE *id_fp, int id):

- This function returns the first letter of the surname of a user with a given identification number from the "second 1.txt" file.
- The file pointer "id_fp" is used to open the file and read the data.
- A loop checks the user identification number until the end of the file.
- Once the identification number of the desired user is found, the first letter of their surname is returned.

3-) get_id_fi(FILE *id_fp, char first_initial):

- This function prints the identification numbers of users whose names start with a specific letter from the "second 1.txt" file.
- The file pointer "id_fp" is used to open the file and read the data.
- A loop checks each line for the name and identification number.
- The identification numbers of users with names starting with the specified letter are printed.

4-) get_id_li(FILE *id_fp, char last_initial):

- This function prints the identification numbers of users whose surnames start with a specific letter from the "second 1.txt" file.
- The file pointer "id_fp" is used to open the file and read the data.
- A loop checks each line for the surname and identification number.
- The identification numbers of users with surnames starting with the specified letter are printed.

5-) average_grade(FILE *info_fp, int id):

- This function calculates the average grade of a user with a given identification number from the "first 1.txt" file.
- The file pointer "info_fp" is used to open the file and read the data.
- A loop checks each line for the identification number.
- Once the identification number of the desired user is found, their exam scores are retrieved, and the average is calculated.

User Interface:

- The main function works by calling specific functions based on user selection. A menu is presented to the user, and the desired operation is chosen. After the functions are called, the results are printed to the screen.

Example Usage:

- The program operates with data read from files named "second 1.txt" and "first 1.txt". Users can process user information by selecting specific functions from the menu and obtain the desired results.

In this way, the program processes user information to accomplish specific tasks and presents the results to the user.

```

1 #include <stdio.h>
2
3 char first_initial(FILE *id_fp, int id) {
4     char firstInitial;
5     int currentId = 0;
6
7     id_fp = fopen("second 1.txt", "r");
8     if (id_fp == NULL) {
9         printf("Error!");
10        return -1;
11    }
12
13    while (fscanf(id_fp, "%d;", &currentId) != EOF) {
14        if (currentId == id) {
15            fscanf(id_fp, " %c", &firstInitial);
16            fclose(id_fp);
17            return firstInitial;
18        } else {
19            while (fgetc(id_fp) != '\n' && !feof(id_fp));
20        }
21    }
22
23    fclose(id_fp);
24    return '\0';
25 }
26
27 char last_initial(FILE *id_fp, int id) {
28     char lastInitial, tempChar;
29     int currentId = 0;
30
31     id_fp = fopen("second 1.txt", "r");
32     if (id_fp == NULL) {
33         printf("Error!");
34         return -1;
35     }
36
37     while (fscanf(id_fp, "%d;", &currentId) != EOF)
38     {
39         if (currentId == id)
40         {
41             while (fscanf(id_fp, "%c", &tempChar) == 1 && tempChar != ';');
42             fscanf(id_fp, " %c", &lastInitial);
43             fclose(id_fp);
44             return lastInitial;
45         }
46         else
47         {
48             while (fgetc(id_fp) != '\n' && !feof(id_fp));
49         }
50     }
51
52     fclose(id_fp);
53     return '\0';
54 }

```

```

56 int get_id_fi(FILE *id_fp, char first_initial)
57 {
58     char currentFirstInitial, tempChar;
59     int idFind = 0;
60
61     id_fp = fopen("second 1.txt", "r");
62     if (id_fp == NULL) {
63         printf("Error!");
64         return -1;
65     }
66
67     while (fscanf(id_fp, "%d;", &idFind) != EOF)
68     {
69         fscanf(id_fp, " %c", &currentFirstInitial);
70
71         while (fscanf(id_fp, "%c", &tempChar) == 1 && tempChar != '\n');
72
73         if (currentFirstInitial == first_initial)
74         {
75             printf("%d\n", idFind);
76         }
77     }
78
79     if (fclose(id_fp) == EOF)
80     {
81         printf("Error!");
82         return -1;
83     }
84
85     return 0;
86 }

```

```

88 int get_id_li(FILE *id_fp, char last_initial) {
89     char currentFirstInitial, currentLastInitial, tempChar;
90     int idFind = 0;
91
92     id_fp = fopen("second 1.txt", "r");
93     if (id_fp == NULL) {
94         printf("Error!");
95         return -1;
96     }
97
98     while (fscanf(id_fp, "%d;", &idFind) != EOF) {
99         fscanf(id_fp, " %c", &currentFirstInitial);
100
101         while (fscanf(id_fp, "%c", &tempChar) == 1 && tempChar != ';');
102
103         fscanf(id_fp, " %c", &currentLastInitial); // Read last initial
104
105         while (fscanf(id_fp, "%c", &tempChar) == 1 && tempChar != '\n');
106
107         if (currentLastInitial == last_initial) {
108             printf("%d\n", idFind);
109         }
110     }
111
112     if (fclose(id_fp) == EOF)
113     {
114         printf("Error!");
115         return -1;
116     }
117
118     return 0;
119 }

```

```

121 int average_grade(FILE *info_fp, int id) {
122     int idFind = 0, midterm, midterm1, midterm2, final;
123
124     info_fp = fopen("first 1.txt", "r");
125     if (info_fp == NULL)
126     {
127         printf("Error!");
128         return -1;
129     }
130
131     while (idFind != id)
132     {
133         fscanf(info_fp, "%d", &idFind);
134     }
135
136     fscanf(info_fp, "%d", &midterm1);
137     fscanf(info_fp, "%d", &midterm2);
138     fscanf(info_fp, "%d", &final);
139
140     if (final > 100)
141     {
142         midterm = midterm1;
143         final = midterm2;
144     }
145     else
146     {
147         midterm = (midterm1 + midterm2) / 2;
148     }
149     return (midterm + final) / 2;
150 }

```

```

152 int main() {
153     printf("----- utility functions demo\n");
154     char firstInitial = first_initial(NULL, 220015014);
155     char lastInitial = last_initial(NULL, 220015014);
156     printf("First initial of given example id (220015014): %c\n", firstInitial);
157     printf("Last initial of given example id (220015014): %c\n", lastInitial);
158     printf("ID's of people whose name starts with given example letter ('m'):\n");
159     get_id_fi(NULL, 'm');
160     printf("\n");
161     printf("ID's of people whose surname starts with given example letter ('m'):\n");
162     get_id_li(NULL, 'm');
163     printf("\n");
164     printf("Average grade of given example id (220015014): %d\n", average_grade(NULL, 210015012));
165     printf("----- end of utility functions demo\n\n");
166     char selection = '0';
167     while (selection != 'e') {
168         char functionchar = '0';
169         int functionint = 0;
170         printf("p: Print all the users pass or fail.\n");
171         printf("n: Print only the user with a specific first initial.\n");
172         printf("g: Calculate the GPA of a given student.\n");
173         printf("c: Print whether each user passed or failed the same class.\n");
174         printf("t: Print the number of classes for the instructor.\n");
175         printf("d: Print the department of all persons according to the role.\n");
176         printf("l: Print the course_id of a given student.\n");
177         printf("e: Quit the program.\n");
178         printf("Please select: ");
179         scanf("%c", &selection);
180         while (getchar() != '\n');

```

```

180         while (getchar() != '\n');
181         switch(selection) {
182             case 'p':
183                 break;
184             case 'n':
185                 printf("Please enter the first initial: ");
186                 break;
187             case 'g':
188                 printf("Please enter the ID: ");
189                 break;
190             case 'c':
191                 printf("Please enter the course ID: ");
192                 break;
193             case 't':
194                 printf("Please enter the ID: ");
195                 break;
196             case 'd':
197                 printf("Please enter the role initial: ");
198                 break;
199             case 'l':
200                 printf("Please enter the ID: ");
201                 break;
202             case 'e':
203                 printf("Exiting program...\n\n");
204                 break;
205             default:
206                 printf("Invalid selection!\n");
207         }
208     }
209
210     return 0;
211 }

```

```

First initial of given example id (220015014): c
Last initial of given example id (220015014): r
ID's of people whose name starts with given example letter ('m'):
220015004
220015005
220015015
220015016
210015007
210015013
195001004
195001008

ID's of people whose surname starts with given example letter ('m'):
220015001
220015011
220015018
210015004
210015010
210015016

```

```
Average grade of given example id (220015014): 66
----- end of utility functions demo

p: Print all the users pass or fail.
n: Print only the user with a specific first initial.
g: Calculate the GPA of a given student.
c: Print whether each user passed or failed the same class.
t: Print the number of classes for the instructor.
d: Print the department of all persons according to the role.
l: Print the course_id of a given student.
e: Quit the program.
Please select: q
Invalid selection!
p: Print all the users pass or fail.
n: Print only the user with a specific first initial.
g: Calculate the GPA of a given student.
c: Print whether each user passed or failed the same class.
t: Print the number of classes for the instructor.
d: Print the department of all persons according to the role.
l: Print the course_id of a given student.
e: Quit the program.
Please select: |
```

Youtube Link :

https://youtu.be/vS_BAJnSnlw