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.

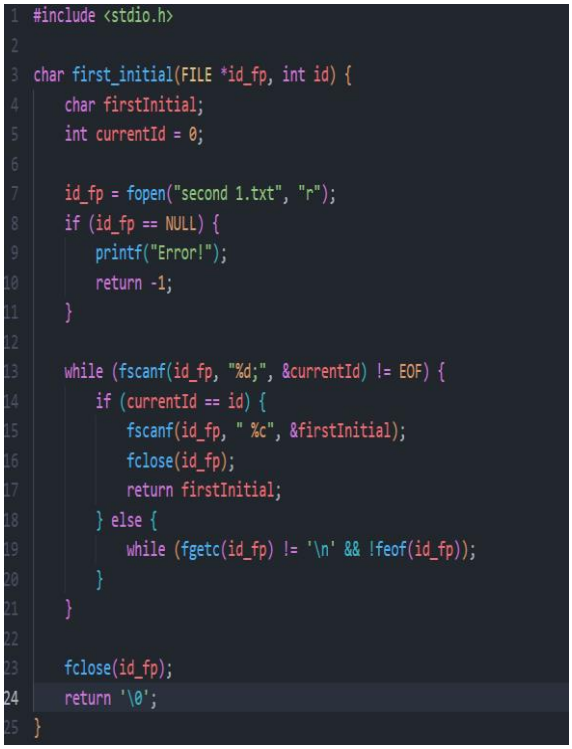
User Interface:

- 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.



```c
#include <stdio.h>

char first_initial(FILE *id_fp, int id) {
    char firstInitial;
    int currentId = 0;

    id_fp = fopen("second 1.txt", "r");
    if (id_fp == NULL) {
        printf("Error!");
        return -1;
    }

    while (fscanf(id_fp, "%d;", &currentId) != EOF) {
        if (currentId == id) {
            fscanf(id_fp, " %c", &firstInitial);
            fclose(id_fp);
            return firstInitial;
        } else {
            while (fgetc(id_fp) != '\n' && !feof(id_fp));
        }
    }

    fclose(id_fp);
    return '\0';
}
```

```c
char last_initial(FILE *id_fp, int id) {
    char lastInitial, tempChar;
    int currentId = 0;

    id_fp = fopen("second 1.txt", "r");
    if (id_fp == NULL) {
        printf("Error!");
        return -1;
    }

    while (fscanf(id_fp, "%d;", &currentId) != EOF)
    {
        if (currentId == id)
        {
            while (fscanf(id_fp, "%c", &tempChar) == 1 && tempChar != ';');
            fscanf(id_fp, " %c", &lastInitial);
            fclose(id_fp);
            return lastInitial;
        }
        else
        {
            while (fgetc(id_fp) != '\n' && !feof(id_fp));
        }
    }

    fclose(id_fp);
    return '\0';
}
```

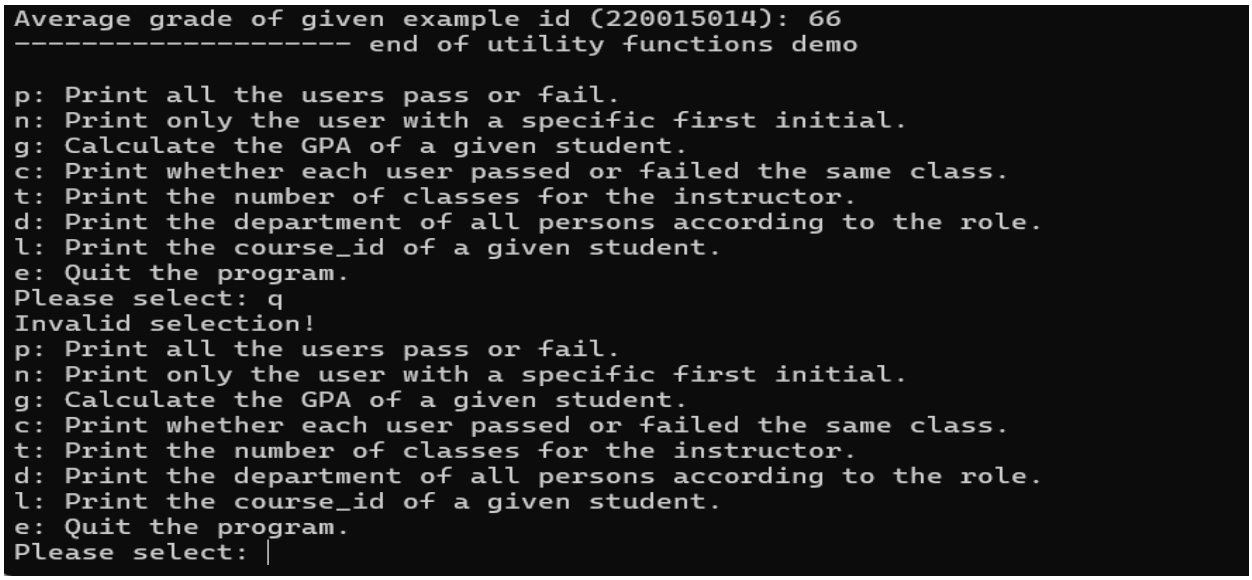
```c
int get_id_fi(FILE *id_fp, char first_initial)
{
    char currentFirstInitial, tempChar;
    int idFind = 0;

    id_fp = fopen("second 1.txt", "r");
    if (id_fp == NULL) {
        printf("Error!");
        return -1;
    }

    while (fscanf(id_fp, "%d;", &idFind) != EOF)
    {
        fscanf(id_fp, " %c", &currentFirstInitial);

        while (fscanf(id_fp, "%c", &tempChar) == 1 && tempChar != '\n');

        if (currentFirstInitial == first_initial)
        {
            printf("%d\n", idFind);
        }
    }

    if (fclose(id_fp) == EOF)
    {
        printf("Error!");
        return -1;
    }

    return 0;
}
```

```c
int get_id_li(FILE *id_fp, char last_initial) {
    char currentFirstInitial, currentLastInitial, tempChar;
    int idFind = 0;

    id_fp = fopen("second 1.txt", "r");
    if (id_fp == NULL) {
        printf("Error!");
        return -1;
    }

    while (fscanf(id_fp, "%d;", &idFind) != EOF) {
        fscanf(id_fp, " %c", &currentFirstInitial);

        while (fscanf(id_fp, "%c", &tempChar) == 1 && tempChar != ';');

        fscanf(id_fp, " %c", &currentLastInitial); // Read last initial

        while (fscanf(id_fp, "%c", &tempChar) == 1 && tempChar != '\n');

        if (currentLastInitial == last_initial) {
            printf("%d\n", idFind);
        }
    }

    if (fclose(id_fp) == EOF)
    {
        printf("Error!");
        return -1;
    }

    return 0;
}
```

```c
int average_grade(FILE *info_fp, int id) {
    int idFind = 0, midterm,midterm1, midterm2, final;

    info_fp = fopen("first 1.txt", "r");
    if (info_fp == NULL)
    {
        printf("Error!");
        return -1;
    }

    while (idFind != id)
    {
        fscanf(info_fp, "%d;", &idFind);
    }

    fscanf(info_fp, "%d;", &midterm1);
    fscanf(info_fp, "%d;", &midterm2);
    fscanf(info_fp, "%d;", &final);

    if (final > 100)
    {
        midterm = midterm1;
        final = midterm2;
    }
    else
    {
        midterm = (midterm1 + midterm2) / 2;
    }
    return (midterm + final) / 2;
}
```

```c
int main() {
    printf("------------------ utility functions demo\n");
    char firstInitial = first_initial(NULL, 220015014);
    char lastInitial = last_initial(NULL, 220015014);
    printf("First initial of given example id (220015014): %c\n", firstInitial);
    printf("Last initial of given example id (220015014): %c\n", lastInitial);
    printf("ID's of people whose name starts with given example letter ('m'):\n");
    get_id_fi(NULL, 'm');
    printf("\n");
    printf("ID's of people whose surname starts with given example letter ('m'):\n");
    get_id_li(NULL, 'm');
    printf("\n");
    printf("Average grade of given example id (220015014): %d\n", average_grade(NULL, 210015012));
    printf("------------------ end of utility functions demo\n\n");
    char selection ='0';
    while (selection != 'e') {
        char functionchar = '0';
        int functionint = 0;
        printf("p: Print all the users pass or fail.\n");
        printf("n: Print only the user with a specific first initial.\n");
        printf("g: Calculate the GPA of a given student.\n");
        printf("c: Print whether each user passed or failed the same class.\n");
        printf("t: Print the number of classes for the instructor.\n");
        printf("d: Print the department of all persons according to the role.\n");
        printf("l: Print the course_id of a given student.\n");
        printf("e: Quit the program.\n");
        printf("Please select: ");
        scanf("%c", &selection);
        while (getchar() != '\n');
        switch(selection) {
            case 'p':
                break;
            case 'n':
                printf("Please enter the first initial: ");
                break;
            case 'g':
                printf("Please enter the ID: ");
                break;
            case 'c':
                printf("Please enter the course ID: ");
                break;
            case 't':
                printf("Please enter the ID: ");
                break;
            case 'd':
                printf("Please enter the role initial: ");
                break;
            case 'l':
                printf("Please enter the ID: ");
                break;
            case 'e':
                printf("Exiting program...\n\n");
                break;
            default:
                printf("Invalid selection!\n");
        }
    }

    return 0;
}
```

```
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
```



Youtube Link :

https://youtu.be/vS_BAJnSnlw