Header file

```c
#ifndef OHMYLIB_H   /* check if not define ohmylib.h */
#define OHMYLIB_H   /* define ohmylib.h */
#include    <stdio.h>
#include    <stdlib.h>
#include    <string.h>
#include    <ctype.h>

#ifdef _WIN32
# define clear()    system("cls")
#else
# define clear()    system("clear")
#endif

#define FILENAME    "data.txt"

typedef struct  s_student_data
{
    char    student_id[11];
    char    subject[11][31];
    float   grade[11];
    float   credit[11];
}   t_student_data;

typedef struct  s_avg_data
{
    char    student_id[11];
    float   grade;
    float   credit;
}   t_avg_data;

int     calculate(void);
int     select_menu(void);
int     remove_data(char id[11]);
int     check_student_id(char *dest);
int     check_subject_name(char *dest);
int     check_subject_grade(float *dest);
int     check_subject_credit(float *dest);
int     load_data(char *raw_data, char *user_id, t_student_data *data);
int     find_data(char *filename, char *user_id, t_student_data *data);
void    add_data();
void    print_welcome();
void    print_menu(void);
void    print_header(char *str);
void    show_grade(t_student_data *data);
void    reset_data(t_student_data *data);
void    print_color(char *str, unsigned char color);
void    write_file(char *filename, t_student_data stu_data);
void    calculate_grade(t_avg_data *dest, t_student_data src);

#endif

```

Main.c

```c
/************************************************************/
/*                                                          */
/*  ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒  */
/*                                                          */
/************************************************************/

#include    "../includes/ohmylib.h"     /* include library */

int main(void)
{
    int choice;

    clear();
    print_welcome();
    while (1)
    {
        while (!(choice = select_menu()))
        {
            clear();
            print_color("\nError, please Enter number 1 to 4 only!\n\n", 198);
        }
        switch (choice)
        {
            case 1:
                add_data();
                break;

            case 2:
                calculate();
                break;

            case 3:
                remove_data("");
                break;

            case 4:
                print_color("\nExit Program!\n\n", 43);
                exit(0);

            default:
                break;
        }
    }
    return(0);
}

```

Interface.c

```c
#include    "../includes/ohmylib.h"

void    print_color(char *str, unsigned char color)
{
    printf("\x1b[38;5;%dm%s\e[0m", color, str);
}

void    print_header(char *str)
{
    char    text[66];
    int     color_arr[6] = {69, 105 ,141, 177 ,213};

    bzero(text, 66);
    text[64] = '\n';
    memset(text, '*', 64);
    print_color(text, color_arr[0]);

    bzero(text, 63);
    memset(text, ' ', 64);
    text[0] = '*';
    text[63] = '*';
    print_color(text, color_arr[1]);

    strncpy(text + (64 - strlen(str)) / 2, str, strlen(str));
    print_color(text, color_arr[2]);

    memset(text, ' ', 63);
    text[0] = '*';
    text[63] = '*';
    print_color(text, color_arr[3]);

    memset(text, '*', 63);
    print_color(text, color_arr[4]);
}

int check_student_id(char *dest)
{
    char    id[31];
    size_t  i;

    printf("Enter Student ID : ");
    scanf("%s", id);
    if (strlen(id) != 10)
        return (0);
    for (i = 0; i < 10; i++)
        if (!isdigit(id[i]))
            return (0);
    strcpy(dest, id);
    return (1);
}
```

```
51      clear();
52      if (answer == 'n')
53      {
54          print_header(show_id);
55          puts("");
56          show_grade(&stu_data);
57          remove_data(stu_data.student_id);
58          write_file(FILENAME, stu_data);
59          print_color("\nAdd Data: Add Grade Complete!\n\n", 35);
60      }
61  }
62
63  void calculate_grade(t_avg_data *dest, t_student_data src)
64  {
65      size_t  i;
66      float   total;
67      float   credit;
68
69      credit = 0;
70      total = 0;
71      i = 0;
72      while (i < 10 && src.subject[i][0] != 0)
73      {
74          total += (src.grade[i] * src.credit[i]);
75          credit += src.credit[i];
76          i++;
77      }
78      strcpy(dest->student_id, src.student_id);
79      dest->grade = total / credit;
80      dest->credit = credit;
81  }
82
83  int calculate(void)
84  {
85      t_student_data  data;
86      t_avg_data      data_avg;
87      char            show_id[38]; bzero(show_id, 38);
88
89      clear();
90      print_header("Calculate GPA");
91      print_color("\nPlease Enter Student ID that you want to calculate.\n\n", 43);
92      reset_data(&data);
93      while (!check_student_id(data.student_id))
94          print_color("Error, please Enter Student ID! (10 digit)\n", 220);
95      sprintf(show_id, "Calculate GPA Student ID : %s", data.student_id);
96      if (find_data(FILENAME, data.student_id, &data))
97      {
98          clear();
99          print_header(show_id);
100         puts("");
```

```c
        puts("");
        show_grade(&data);
        calculate_grade(&data_avg, data);
        puts("");
        printf("Average Grade\t= %.2f\n",data_avg.grade);
        printf("Total credit\t= %.2f\n",data_avg.credit);
        puts("");
        return (1);
    }
    else
    {
        print_color("\nCalculate GPA: ID Not Found!!, Please Add Data before!\n\n", 198);
        return (0);
    }
}

int remove_data(char *id)
{
    t_student_data  temp;
    int             status = 0;
    FILE            *new_file, *old_file;
    char            buff_file[] = "buff.txt";
    char            buff[sizeof(t_student_data) + 10 + 1];
    if (!*id) /* *id == id[0] */
    {   clear();
        print_header("Remove Data");
        print_color("\nPlease Enter Student ID that you want to Delete data.\n\n", 43);

        id = (char *) malloc (sizeof(char *) * 11);
        if (!id)
            return (0);
        while (!check_student_id(id))
            print_color("Error, please Enter Student ID! (10 digit)\n", 220);}
    else
        status = 1;
    if (find_data(FILENAME, id, &temp))
    {
        rename(FILENAME, buff_file);
        old_file = fopen(buff_file, "r");
        new_file = fopen(FILENAME, "w+");
        if (!old_file || !new_file)
        {
            fclose(old_file);
            fclose(new_file);
            return (0);
        }
        while (!feof(old_file))
        {
            fscanf(old_file, "%s\n", buff);
            if (strncmp(id, buff, 10))
                fprintf(new_file, "%s\n", buff);
        }
        fclose(old_file);
        fclose(new_file);
        remove(buff_file);
        if (!status)
            print_color("\nRemove Data: Remove Old Data Complete!\n\n", 43);
    }
    else
        if (!status)
            print_color("\nRemove Data: Old Data not found, You may not delete.\n\n", 191);
    return (1);
}

```

Utility.c

```c
#include    "../includes/ohmylib.h"

void    print_color(char *str, unsigned char color)
{
    printf("\x1b[38;5;%dm%s\e[0m", color, str);
}

void    print_header(char *str)
{
    char    text[66];
    int     color_arr[6] = {69, 105 ,141, 177 ,213};

    bzero(text, 66);
    text[64] = '\n';
    memset(text, '*', 64);
    print_color(text, color_arr[0]);

    bzero(text, 63);
    memset(text, ' ', 64);
    text[0] = '*';
    text[63] = '*';
    print_color(text, color_arr[1]);

    strncpy(text + (64 - strlen(str)) / 2, str, strlen(str));
    print_color(text, color_arr[2]);

    memset(text, ' ', 63);
    text[0] = '*';
    text[63] = '*';
    print_color(text, color_arr[3]);

    memset(text, '*', 63);
    print_color(text, color_arr[4]);
}

int check_student_id(char *dest)
{
    char    id[31];
    size_t  i;

    printf("Enter Student ID : ");
    scanf("%s", id);
    if (strlen(id) != 10)
        return (0);
    for (i = 0; i < 10; i++)
        if (!isdigit(id[i]))
            return (0);
    strcpy(dest, id);
    return (1);
}

int check_subject_name(char *dest)
{
    char    buffer[51];

    bzero(buffer, 51);
    printf("Enter Subject Name :\t");
    scanf("%s", buffer);
    if (strlen(buffer) > 30)
        return (0);
    else
    {
        strcpy(dest, buffer);
        return (1);
    }
}

```

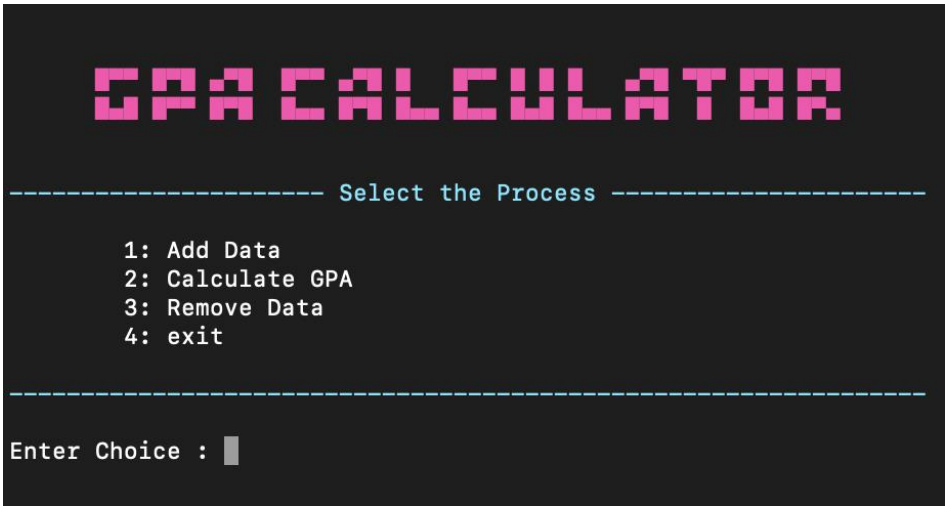
```c
int check_subject_grade(float *dest)
{
    char    src[21];
    size_t  i;

    i = 0;
    printf("Enter Subject Grade :\t");
    scanf("%s", src);
    for (i = 0; i < 21 && src[i]; i++)
        if (!(isdigit(src[i]) || src[i] == '.'))
            return (0);
    if (*src)
    {
        *dest = atof(src);
        if (0 <= *dest && *dest <= 4)
            return (1);
        else
            *dest = 0;
    }
    return (0);
}


int check_subject_credit(float *dest)
{
    char    src[21];
    size_t  i;

    i = 0;
    printf("Enter Subject Credit :\t");
    scanf("%s", src);
    for (i = 0; i < 21 && src[i]; i++)
        if (!(isdigit(src[i]) || src[i] == '.'))
            return (0);
    if (*src)
    {
        *dest = atof(src);
        if (0 <= *dest && *dest <= 10)
            return (1);
        else
            *dest = 0;
    }
    return (0);
}

void    show_grade(t_student_data *data)
{
    for(size_t i = 0; i < 10 && *data->subject[i]; i++)
    {
        printf("------------------------ SUBJECT %d ------------------------\n", (int) i + 1);
        printf("Subject Name :\t\t%s\n", data->subject[i]);
        printf("Subject Grade :\t\t%.2f\n", data->grade[i]);
        printf("Subject Credit :\t%.2f\n\n", data->credit[i]);
    }
    printf("----------------------------------------------------------------\n");
}

void    print_menu(void)
{
    print_color("---------------------- Select the Process ----------------------\n", 81);
    printf("\n");
    printf("\t1: Add Data\n");
    printf("\t2: Calculate GPA\n");
    printf("\t3: Remove Data\n");
    printf("\t4: exit\n");
    print_color("\n----------------------------------------------------------------\n", 81);
    printf("\n");
}

```

```c
int select_menu(void)
{
    char    str[31];
    int     c;

    c = 0;
    bzero(str, 31);
    print_menu();
    printf("Enter Choice : ");
    scanf("%s", str);
    for (size_t i = 0; i < 31 && str[i]; i++)
        if (!isdigit(str[i]))
            return (0);
    c = atoi(str);
    if (1 <= c && c <= 4)
        return (c);
    return(0);
}

#ifndef _WIN32
void    print_welcome()
{
    char LOADING[] = "\n\n\
                █░░ █▀█ ▄▀█ █▀▄ █ █▄░█ █▀▀ ░ ░ ░\n\
                █▄▄ █▄█ █▀█ █▄▀ █ █░▀█ █▄█ ▄ ▄ ▄\n\n\n";
    char GPA_CALCULATOR[] = "\n\n\
     █▀▀ █▀█ ▄▀█   █▀▀ ▄▀█ █░░ █▀▀ █░█ █░░ ▄▀█ ▀█▀ █▀█ █▀█\n\
     █▄█ █▀▀ █▀█   █▄▄ █▀█ █▄▄ █▄▄ █▄█ █▄▄ █▀█ ░█░ █▄█ █▀▄\n\n\n";

    clear();
    print_color(LOADING, 214);
    system("sleep 0.6");
    clear();
    print_color(GPA_CALCULATOR, 162);
}
#else
void print_welcome()
{
    print_header("GPA CALCULATOR");
}
#endif
```

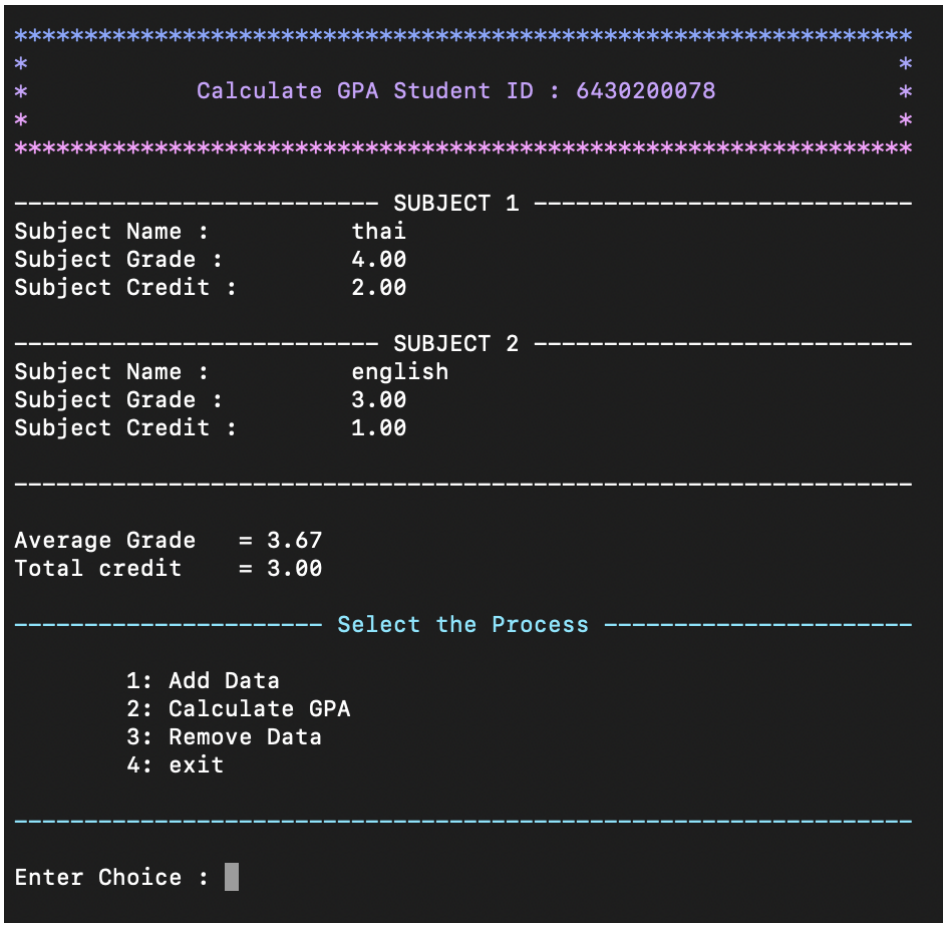
File.c

```c
#include    "../includes/ohmylib.h"

void write_file(char *filename, t_student_data stu_data)
{
    FILE *fp_add_grade;
    fp_add_grade=fopen(filename,"a+");
    fprintf(fp_add_grade,"%s",stu_data.student_id);
    fprintf(fp_add_grade,",");
    for(size_t i = 0; i < 10; i++)
    {
        if(*stu_data.subject[i])
        {
            fprintf(fp_add_grade,"%s",stu_data.subject[i]);
            fprintf(fp_add_grade,",");
            fprintf(fp_add_grade,"%.2f",stu_data.grade[i]);
            fprintf(fp_add_grade,",");
            fprintf(fp_add_grade,"%.2f",stu_data.credit[i]);
        }
        else
            fprintf(fp_add_grade,"-,-,-");
        if (i < 9)
            fprintf(fp_add_grade,",");
    }
    fprintf(fp_add_grade,"\n");
    fclose(fp_add_grade);
}

int load_data(char *raw_data, char *user_id, t_student_data *data)
{
    char    *ptr_str;
    char    *token;

    ptr_str = raw_data;
    token = strtok_r(ptr_str, ",", &ptr_str);
    if (!strcmp(user_id, token))
    {
        strcpy(data->student_id, token);
        for (size_t i = 0; token && i < 10; i++)
        {
            token = strtok_r(ptr_str, ",", &ptr_str);
            if (*token == '-')
                break ;
            strcpy(data->subject[i], token);
            token = strtok_r(ptr_str, ",", &ptr_str);
            data->grade[i] = atof(token);
            token = strtok_r(ptr_str, ",", &ptr_str);
            data->credit[i] = atof(token);
        }
        return (1);
    }
    else
        return (0);
}

int find_data(char *filename, char *user_id, t_student_data *data)
{
    char    buff[sizeof(t_student_data) + 10 + 1];
    int     found;
    FILE    *file;

    found = 0;
    file = fopen(filename, "r");
    if (!file)
        return (0);
    while (!feof(file) && !found)
    {
        fscanf(file, "%s\n", buff);
        if (!*buff)
            return (found);
        else
            found = load_data(buff, user_id, data);
    }
    fclose(file);
    return (found);
}
```

ตัวอย่างผลรัน


GPA CALCULATOR
---------- Select the Process ----------
1: Add Data
2: Calculate GPA
3: Remove Data
4: exit

Enter Choice :


****
* *
* Calculate GPA Student ID : 6430200078 *
* *
****
---------- SUBJECT 1 ----------
Subject Name : thai
Subject Grade : 4.00
Subject Credit : 2.00
---------- SUBJECT 2 ----------
Subject Name : english
Subject Grade : 3.00
Subject Credit : 1.00

Average Grade = 3.67
Total credit = 3.00
---------- Select the Process ----------
1: Add Data
2: Calculate GPA
3: Remove Data
4: exit

Enter Choice :

```
**************************************************************************
*                                                                        *
*                            Add Data                                    *
*                                                                        *
**************************************************************************

Please Enter Student ID that you want to add data.

Enter Student ID : 1111111111111
Error, please Enter Student ID! (10 digit)
Enter Student ID : aaaaaaaaaaaa
Error, please Enter Student ID! (10 digit)
Enter Student ID : a5553444444
Error, please Enter Student ID! (10 digit)
Enter Student ID : 255555555
Error, please Enter Student ID! (10 digit)
Enter Student ID : 6430200078
```

```
**************************************************************************
*                                                                        *
*                Add Data Student ID : 6430200078                        *
*                                                                        *
**************************************************************************

Enter Subject Name :    eng
Enter Subject Grade :   -1
Error, please Enter Grade(float) only! (0.00 to 4.00)
Enter Subject Grade :   98
Error, please Enter Grade(float) only! (0.00 to 4.00)
Enter Subject Grade :   10
Error, please Enter Grade(float) only! (0.00 to 4.00)
Enter Subject Grade :   15
Error, please Enter Grade(float) only! (0.00 to 4.00)
Enter Subject Grade :   4
```

รายชื่อสมาชิก

นางสาวจุฬาลักษณ์ หาทอน รหัสนิสิต 6430200078

นายฐปกร อิ่มอักษร รหัสนิสิต 6430200159

นางสาวภคมน ดวงแก้วเลิศ รหัสนิสิต 6430200531

นางสาวศิรประภา ยอดศิริ รหัสนิสิต 6430200671

นายพงศกร ทิพยสมเดช รหัสนิสิต 6430200850

หมู่ปฏิบัติการ 831