```c
/*
 * FIFO.c
 *
 *  Created on: 12 Sept 2022
 *      Author: MAshr
 */
#include"FIFO.h"

FIFO_status FIFO_init(FIFO_buf_t *fifo, element_type *buf, unsigned int length) {

    if (buf == NULL)
        return FIFO_null;

    fifo->base = buf;
    fifo->tail = buf;
    fifo->head = buf;
    fifo->count = 0;
    fifo->length = length;

    return FIFO_no_error;
}

FIFO_status FIFO_enqueue(FIFO_buf_t *fifo, element_type item) {

    if (!fifo->base || !fifo->tail || !fifo->head)
        return FIFO_null;
    if (FIFO_is_full(fifo) == FIFO_full)
        return FIFO_full;

    *(fifo->head) = item;
    if (fifo->head == fifo->base + (fifo->length * sizeof(element_type)))
        fifo->head = fifo->base;
    else
        fifo->head++;

    fifo->count++;
    return FIFO_no_error;
}


FIFO_status FIFO_dequeue(FIFO_buf_t *fifo, element_type *item) {

    if (!fifo->base || !fifo->tail || !fifo->head)
        return FIFO_null;
    if (fifo->count == 0)
        return FIFO_empty;

    *item = *fifo->tail;
    fifo->count--;

    if (fifo->tail == fifo->base + (fifo->length * sizeof(element_type)))
        fifo->tail = fifo->base;
    else
        fifo->tail++;


    return FIFO_no_error;
}

```

```c
FIFO_status FIFO_is_full(FIFO_buf_t *fifo) {

    if (!fifo->base || !fifo->tail || !fifo->head)
        return FIFO_null;
    if (fifo->count == fifo->length)
        return FIFO_full;
    else return FIFO_no_error;
}

void FIFO_print(FIFO_buf_t *fifo) {

    int i;
    element_type *ptrtemp;

    if (fifo->count == 0)
    {
        Dprintf("FIFO is empty\n");
    }
    else {
        ptrtemp = fifo->tail;
        Dprintf("\n================ print =============== \n");
        for (i = 0; i < fifo->count; i++) {
           // Dprintf("\t (%x) \n", *ptrtemp);
            ptrPrinterFnc(ptrtemp);
            ptrtemp++;
        }
    }
}

FIFO_status FIFO_delete_element(FIFO_buf_t* fifo,element_type* item)
{
    if (!fifo->base || !fifo->tail || !fifo->head)
        return FIFO_null;
    if (fifo->count == 0)
        return FIFO_empty;
int i= 0 ;

    while (item < fifo->head)
    {
        *item=*(item+1);

        if (item == fifo->base + (fifo->length * sizeof(element_type)))
            item = fifo->base;
        else
            item++;
    }
    fifo->count-- ;

    if (fifo->head == fifo->base)
        fifo->head = fifo->base + (fifo->length * sizeof (element_type));
    else
        fifo->head--;
    return FIFO_no_error ;
}

```

```c
/*
 * FIFO.h
 *
 *  Created on: 12 Sept 2022
 *      Author: MAshr
 */

#ifndef FIFO_H_
#define FIFO_H_

#include<stdio.h>
#include<stdint.h>

#include "FIFO_config.h"

#define Dprintf(...) printf(__VA_ARGS__);fflush(stdout);fflush(stdin)

//create buffer

element_type uart_buff[buffSize];

typedef struct {
    unsigned int length ;
    unsigned int count ;
    element_type *head ;
    element_type *base ;
    element_type *tail ;
}FIFO_buf_t;

typedef enum {
    FIFO_no_error ,
    FIFO_full,
    FIFO_empty,
    FIFO_null
}FIFO_status;


// API's

FIFO_status FIFO_init(FIFO_buf_t* fifo , element_type* buf, unsigned int length) ;
FIFO_status FIFO_enqueue(FIFO_buf_t* fifo , element_type item) ;
FIFO_status FIFO_dequeue(FIFO_buf_t* fifo , element_type* item) ;
FIFO_status FIFO_is_full(FIFO_buf_t* fifo) ;
void FIFO_print(FIFO_buf_t* fifo) ;
FIFO_status FIFO_delete_element(FIFO_buf_t* fifo, element_type* item);


#endif /* FIFO_H_ */

```

```c
/*
 * main.c
 *
 *  Created on: 12 Sept 2022
 *      Author: MAshr
 */
#include "Management_System.h"
#include "stdlib.h"
int main(void) {
    int choice;
    systemCreate();
    while (1) {
        systemPrintMenu();
        scanf("%d",&choice);
        switch (choice) {
            case 1:
                systemAddStudent_manually();
                break;
            case 2:
                loadDataFile();
                break;
            case 3:
                systemFindStudentRoll();
                break;
            case 4:
                systemFindStudentName();
                break;
            case 5:
                systemFindStudentCourseID();
                break;
            case 6:
                systemStatistics();
                break;
            case 7:
                systemDeleteStudent();
                break;
            case 8:
                systemUpdateStudent();
                break;
            case 9:
                systemPrintAll();
                break;
            case 10:
                systemUpdateStudentFile();
                break;
            case 11:
                exit(1);
        }
    }
}

```

```
1 1 Mina Karam 2.80 1 2 3 4 5
2 3 Leila Mustafa 3.40 54 12 55 81 64
3 4 Koky Samy 3.60 5 21 55 92 64
4 5 Esraa Hegazy 3.60 54 12 55 81 64
5 2 Mahmoud Ashraf 4.00 15 1 20 3 40
6
```

```c
//
// Created by MAshr on 22/09/2022.
//

#ifndef STUDENT_MANAGEMENT_SYSTEM_FIFO_CONFIG_H
#define STUDENT_MANAGEMENT_SYSTEM_FIFO_CONFIG_H

#include "Management_system_types.h"

// USER configuration
//select  element type
//fifo data types
#define element_type studentInfo_t
#define buffSize (unsigned int)50
extern systemStatus_t systemPrintStudent(studentInfo_t *);
extern systemStatus_t (*ptrPrinterFnc)(studentInfo_t*);
#endif //STUDENT_MANAGEMENT_SYSTEM_FIFO_CONFIG_H

```

```
cmake_minimum_required(VERSION 3.23)
project(Student_Management_System C)

set(CMAKE_C_STANDARD 11)

include_directories(.)

add_executable(Student_Management_System
        FIFO.c
        FIFO.h
        main.c
        Management_System.h
        FIFO_config.h
        Management_system_types.h
        Management_System.c
        )

```

```
1 1 Ahmed Gamal 2.8 1 2 3 4 5
2 1 Mohamed Fahmy 3.3 5 21 73 92 36
3 3 Mahmoud Mustafa 3.4 54 12 55 81 64
4 4 Ahmed Samy 3.6 5 21 55 92 64
5 5 Islam Hilal 3.6 54 12 55 81 64
```

```c
//
// Created by MAshr on 23/09/2022.
//
/*********************************************************************************************************************
 *  FILE DESCRIPTION
 *  -------------------------------------------------------------------------------------------------------------------*/
/**        \file  FileName.c
 *        \brief
 *
 *      \details
 *
 *
 ********************************************************************************************************************/

/*********************************************************************************************************************
 *  INCLUDES
 ********************************************************************************************************************/
#include "Management_System.h"
#include <string.h>
#include <stdio.h>
#include <stdlib.h>

/*********************************************************************************************************************
*  LOCAL MACROS CONSTANT\FUNCTION
*********************************************************************************************************************/

/*********************************************************************************************************************
 *  LOCAL DATA
 ********************************************************************************************************************/

/*********************************************************************************************************************
 *  GLOBAL DATA
 ********************************************************************************************************************/
studentInfo_t mainBuff[buffSize];
FIFO_buf_t FIFO_info;

systemStatus_t (*ptrPrinterFnc)(studentInfo_t *);

/*********************************************************************************************************************
 *  LOCAL FUNCTION PROTOTYPES
 ********************************************************************************************************************/
studentInfo_t *systemCatchStudentData();

systemStatus_t checkRollID(int rollID);

systemStatus_t systemPrintStudent(studentInfo_t *);
```

```c

systemStatus_t printUpdateMenu();

systemStatus_t updateStudentRoll(studentInfo_t *Student);

systemStatus_t updateStudentFname(studentInfo_t *Student);

systemStatus_t updateStudentLname(studentInfo_t *Student);

systemStatus_t updateStudentGPA(studentInfo_t *Student);

systemStatus_t updateStudentCourses(studentInfo_t *Student);

systemStatus_t FIFO_print_file(FIFO_buf_t *fifo);
/*********************************************************************************************************************
 *  LOCAL FUNCTIONS
 *********************************************************************************************************************/

studentInfo_t *systemCatchStudentData() {
    static studentInfo_t temp;
    char str[20];
    Dprintf("Enter Student Roll ID\n");
    scanf("%d", &temp.roll);
    if (!checkRollID(temp.roll)) {
        Dprintf("Enter Student first Name \n");
        gets(str);
        strcpy(temp.firstName, str);
        Dprintf("Enter Student last Name \n");
        gets(str);
        strcpy(temp.lastName, str);
        Dprintf("Enter Student GPA \n");
        scanf("%f", &temp.GPA);
        Dprintf("Enter Student Courses ID\n");
        for (int i = 0; i < 5; i++) {
            Dprintf("==> Enter Course %d ID \n", i + 1);
            scanf("%d", &temp.courseID[i]);
        }
    } else {
        return NULL;
    }
    return &temp;
}

/*-------------------------------------------------------------------------------------------*/
systemStatus_t printUpdateMenu() {
    Dprintf("Choose which item to update : \n");
    Dprintf("1- Roll ID \n");
    Dprintf("2- First name \n");
    Dprintf("3- Last name \n");
    Dprintf("4- GPA \n");
    Dprintf("5- Courses \n");
    Dprintf("6- Back to main menu\n");
}

systemStatus_t systemPrintMenu() {
    char menu[15][50] = {
            "----------------------------------------\n",
```

```c
            " Chose the task that you want to perform \n",
            "1- Add the students data manually\n",
            "2- Load Students data from file \n",
            "3- Find student details by roll number \n",
            "4- Find student details by first name  \n",
            "5- Find student details by course ID \n",
            "6- Find total number of students\n",
            "7- Delete student details by roll number \n",
            "8- Update student details by roll number \n",
            "9- Show all information\n",
            "10- Save data to output file\n",
            "11- Exit the system\n",
            "===> Enter your choice number \n "
    };
    for (int i = 0; i < 15; i++) {
        Dprintf("%s", menu[i]);
    }

    return System_noError;
}

/*-------------------------------------------------------------------------------------*/
systemStatus_t checkRollID(int rollID) {
    int i = 0;
    while (i < FIFO_info.count) {
        if (FIFO_info.tail[i].roll == rollID) {
            Dprintf("-------------------------------------------\n");
            Dprintf("[ERROR] : This Roll ID (%d) already exist \n", rollID);
            Dprintf("-------------------------------------------\n");

            systemPrintStudent(&FIFO_info.tail[i]);
            return System_Error;
        }
        i++;
    }

    return System_noError;
}

/*-------------------------------------------------------------------------------------*/
systemStatus_t systemPrintStudent(studentInfo_t *student) {
    Dprintf("Student Roll Number : %d\n", student->roll);
    Dprintf("First Name : %s\n", student->firstName);
    Dprintf("Last Name : %s\n", student->lastName);
    Dprintf("GPA: %f\n", student->GPA);
    Dprintf("Registered Courses ID : [");
    for (int i = 0; i < 5; i++) {
        Dprintf("  %d  ", student->courseID[i]);
    }
    Dprintf("]\n---------------------------------------\n");
    return System_noError;
}

systemStatus_t updateStudentRoll(studentInfo_t *Student) {
    int nRollID;
    Dprintf("Enter Student (New) Roll ID \n");
    scanf("%d", &nRollID);
```

```c
    if (!checkRollID(nRollID)) {
        Student->roll = nRollID;
    }
}

systemStatus_t updateStudentFname(studentInfo_t *Student) {
    char str[20];
    Dprintf("Enter Student ( New ) first Name \n");
    gets(str);
    strcpy(Student->firstName, str);
}

systemStatus_t updateStudentLname(studentInfo_t *Student) {
    char str[20];
    Dprintf("Enter Student ( New ) last Name \n");
    gets(str);
    strcpy(Student->lastName, str);
}

systemStatus_t updateStudentGPA(studentInfo_t *Student) {
    Dprintf("Enter Student (New) GPA \n");
    scanf("%f", &Student->GPA);
}

systemStatus_t updateStudentCourses(studentInfo_t *Student) {
    Dprintf("Enter Student (New) Courses ID\n");
    for (int i = 0; i < 5; i++) {
        Dprintf("==> Enter Course %d ID \n", i + 1);
        scanf("%d", &Student->courseID[i]);
    }
}

/*********************************************************************************************************
 *  GLOBAL FUNCTIONS
 *********************************************************************************************************/


systemStatus_t systemCreate() {
    FIFO_init(&FIFO_info, mainBuff, buffSize);
    Dprintf("FIFO has been initialized successfully\n");
    Dprintf("=== WELCOME TO STUDENT MANAGEMENT SYSTEM ===\n");
    //systemPrintMenu();
    return System_noError;
}


systemStatus_t systemAddStudent_manually() {
    studentInfo_t *ptrStudent = systemCatchStudentData();
    if (ptrStudent) {
        FIFO_enqueue(&FIFO_info, *ptrStudent);
        Dprintf("[INFO] student roll %d has been added successfully\n", ptrStudent->roll);
        Dprintf("System stored %d student remains : %d student \n", FIFO_info.count, buffSize - FIFO_info.count);
        return System_noError;
    } else {
        return System_Error;
```

```
216     }
217 }
218
219 systemStatus_t systemFindStudentRoll() {
220     int i = 0, id;
221     Dprintf("Enter Roll ID for the student \n");
222     scanf("%d", (int *) &id);
223     while (i < FIFO_info.count) {
224         if (FIFO_info.tail[i].roll == id) {
225             systemPrintStudent(&FIFO_info.tail[i]);
226             return System_noError;
227         }
228         i++;
229     }
230     Dprintf("-----------------------------------------\n");
231     Dprintf("[ERROR] : This ID doesn't exist \n");
232     Dprintf("-----------------------------------------\n");
233     return System_Error;
234
235 }
236
237 systemStatus_t systemFindStudentName() {
238     int i = 0;
239     char name[20];
240     Dprintf("Enter first name for the student \n");
241     gets(name);
242     while (i < FIFO_info.count) {
243         if (!strcmpi(name, FIFO_info.tail[i].firstName)) {
244             systemPrintStudent(&FIFO_info.tail[i]);
245             return System_noError;
246         }
247         i++;
248     }
249     Dprintf("-----------------------------------------\n");
250     Dprintf("[ERROR] : This name doesn't exist \n");
251     Dprintf("-----------------------------------------\n");
252     return System_Error;
253 }
254
255 systemStatus_t systemFindStudentCourseID() {
256     int i = 0, courseID, count = 0;
257     studentInfo_t *ptrStudent = NULL;
258     Dprintf("Enter Course ID \n");
259     scanf("%d", (int *) &courseID);
260     while (i < FIFO_info.count) {
261         for (int j = 0; j < 5; j++) {
262             if (FIFO_info.tail[i].courseID[j] == courseID) {
263                 ptrStudent = &FIFO_info.tail[i];
264                 Dprintf("The students registered this course ID are : \n");
265                 systemPrintStudent(ptrStudent);
266                 count++;
267             }
268         }
269         i++;
270     }
271     if (ptrStudent) {
272         Dprintf("Number of student register Course ID: (%d) are : (%d)\n", courseID,
    count);
273         Dprintf("-----------------------------------------------------\n");
```

```c
        return System_noError;
    } else {
        Dprintf("[INFO] : No student register this course  \n");
        return System_Error;
    }
}

systemStatus_t systemPrintAll() {
    ptrPrinterFnc = (systemStatus_t (*)(studentInfo_t *)) systemPrintStudent;
    FIFO_print(&FIFO_info);
    return System_noError;
}

systemStatus_t systemStatistics() {
    Dprintf("Maximum size of storage is : %d \n", buffSize);
    Dprintf("Number of students have been stored : %d student \n", FIFO_info.count);
    Dprintf("Number of students remain : %d\n", buffSize - FIFO_info.count);
    Dprintf("-----------------------------------------------------\n");
}

systemStatus_t loadDataFile() {
    FILE *infile;
    studentInfo_t tempStudent;
    infile = fopen("StudentData.txt", "r");
    if (infile == NULL) {
        fprintf(stderr, "\nError opening file\n");
        Dprintf("[ERROR] Can't open the file \n");
    }
    fscanf(infile, "%d", &tempStudent.roll);

    while (!feof(infile)) {
        fscanf(infile, "%d", &tempStudent.roll);
        if (!checkRollID(tempStudent.roll)) {
            fscanf(infile, "%s", tempStudent.firstName);
            fscanf(infile, "%s", tempStudent.lastName);
            fscanf(infile, "%f", &tempStudent.GPA);
            for (int i = 0; i < 5; i++) {
                fscanf(infile, "%d", &tempStudent.courseID[i]);
            }
            FIFO_enqueue(&FIFO_info, tempStudent);
            Dprintf("[INFO] student roll %d has been added successfully\n",
    tempStudent.roll);
            Dprintf("[INFO] System stored %d student remains : %d student \n",
    FIFO_info.count,
                    buffSize - FIFO_info.count);
        } else {
            fscanf(infile, "%*[^\n]");
        }
    }
    fclose(infile);
    return System_noError;
}

systemStatus_t systemDeleteStudent() {
    int i = 0, rollID;
    Dprintf("Enter Roll ID for the student \n");
    scanf("%d", &rollID);
    while (i < FIFO_info.count) {
        if (FIFO_info.tail[i].roll == rollID) {
```

```c
            FIFO_delete_element(&FIFO_info, &FIFO_info.tail[i]);
            Dprintf("[INFO] Student roll %d has been deleted successfully \n", rollID
);
            return System_noError;
        }
        i++;
    }
    Dprintf("-------------------------------------------\n");
    Dprintf("[ERROR] : This ID doesn't exist \n");
    Dprintf("-------------------------------------------\n");
    return System_Error;
}

systemStatus_t systemUpdateStudent() {
    int i = 0, rollID, choice;
    Dprintf("Enter Roll ID for the student \n");
    scanf("%d", &rollID);
    while (i < FIFO_info.count) {
        if (FIFO_info.tail[i].roll == rollID) {
            while (1) {
                printUpdateMenu();
                scanf("%d", &choice);
                switch (choice) {
                    case 1:
                        updateStudentRoll(&FIFO_info.tail[i]);
                        break;
                    case 2:
                        updateStudentFname(&FIFO_info.tail[i]);
                        break;
                    case 3:
                        updateStudentLname(&FIFO_info.tail[i]);
                        break;
                    case 4:
                        updateStudentGPA(&FIFO_info.tail[i]);
                        break;
                    case 5:
                        updateStudentCourses(&FIFO_info.tail[i]);
                        break;
                    case 6:
                        break;
                }
                Dprintf("[INFO] Student information has been updated successfully \n"
);
                if (choice == 6)break;
            }
            return System_noError;
        }
        i++;
    }
    Dprintf("-------------------------------------------\n");
    Dprintf("[ERROR] : This ID doesn't exist \n");
    Dprintf("-------------------------------------------\n");
    return System_Error;
}

systemStatus_t systemUpdateStudentFile() {

    FIFO_print_file(&FIFO_info);
    return System_noError;
```

```c

}

systemStatus_t FIFO_print_file(FIFO_buf_t *fifo) {

    int i;
    element_type *ptrtemp;

    if (fifo->count == 0) {
        Dprintf("FIFO is empty\n");
    } else {
        ptrtemp = fifo->tail;
        Dprintf("\n================ printing in file  =============== \n");

        FILE *outfile;
        outfile = fopen("outfile.txt", "w");
        if (outfile == NULL) {
            fprintf(stderr, "\nError opening file\n");
            Dprintf("[ERROR] Can't open the file \n");
        }
        for (i = 0; i < fifo->count; i++) {

            fprintf(outfile, "%d ", ptrtemp->roll);
            fprintf(outfile, "%s ", ptrtemp->firstName);
            fprintf(outfile, "%s ", ptrtemp->lastName);
            fprintf(outfile, "%.2f ", ptrtemp->GPA);
            for (int j = 0; j < 5; j++) {
                fprintf(outfile, "%d ", ptrtemp->courseID[j]);
            }

            fprintf(outfile,"\n");

            if (ptrtemp == fifo->base + (fifo->length * sizeof(element_type)))
                ptrtemp = fifo->base;
            else
                ptrtemp++;

        }
        Dprintf("[INFO] Data saved to file successfully \n") ;
        fclose(outfile);
    }
}


/*********************************************************************************************************************
 *  END OF FILE: Management_System.c
 ********************************************************************************************************************/

```

```c
//
// Created by MAshr on 22/09/2022.
//

#ifndef STUDENT_MANAGEMENT_SYSTEM_MANAGEMENT_SYSTEM_H
#define STUDENT_MANAGEMENT_SYSTEM_MANAGEMENT_SYSTEM_H

/*******************************************************************************
 ********************************

 *  FILE DESCRIPTION
 *  -----------------------------------------------------------------------------
 ------------------------------
 *         File:  <Management_system.h>
 *       Module:  -
 *
 *  Description:  <           >
 *
 ******************************************************************************
 ********************************/


/*******************************************************************************
 ********************************
 * INCLUDES
 ******************************************************************************
 ********************************/
#include "FIFO.h"
#include "Management_system_types.h"
/*******************************************************************************
 ********************************
 *  GLOBAL CONSTANT MACROS
 ******************************************************************************
 ********************************/


/*******************************************************************************
 ********************************
 *  GLOBAL FUNCTION MACROS
 ******************************************************************************
 ********************************/


/*******************************************************************************
 ********************************
 *  GLOBAL DATA TYPES AND STRUCTURES
 ******************************************************************************
 ********************************/


/*******************************************************************************
 ********************************
 *  GLOBAL DATA PROTOTYPES
 ******************************************************************************
 ********************************/
extern studentInfo_t mainBuff[buffSize];
extern FIFO_buf_t FIFO_info;

/*******************************************************************************
```

```
46 ********************************
47  *  GLOBAL FUNCTION PROTOTYPES
48  ********************************************************************************************
   ********************************/
49 systemStatus_t systemCreate();
50 systemStatus_t systemPrintMenu();
51 systemStatus_t loadDataFile();
52 systemStatus_t systemAddStudent_manually();
53 systemStatus_t systemFindStudentRoll();
54 systemStatus_t systemFindStudentName();
55 systemStatus_t systemFindStudentCourseID();
56 systemStatus_t systemStatistics();
57 systemStatus_t systemDeleteStudent();
58 systemStatus_t systemUpdateStudent();
59 systemStatus_t systemPrintAll();
60 systemStatus_t systemUpdateStudentFile();
61 /********************************************************************************************
   ********************************
62  *  END OF FILE: Std_Types.h
63  ********************************************************************************************
   ********************************/
64
65 #endif //STUDENT_MANAGEMENT_SYSTEM_MANAGEMENT_SYSTEM_H
66
```

```c
//
// Created by MAshr on 22/09/2022.
//

#ifndef STUDENT_MANAGEMENT_SYSTEM_MANAGEMENT_SYSTEM_TYPES_H
#define STUDENT_MANAGEMENT_SYSTEM_MANAGEMENT_SYSTEM_TYPES_H

/*******************************************************************************
 *  GLOBAL DATA PROTOTYPES
 *******************************************************************************/

typedef struct {
    int roll ;
    char firstName[20];
    char lastName[20];
    float GPA ;
    int courseID[5];
}studentInfo_t;

typedef enum {
    System_noError,
    System_Error

}systemStatus_t;
#endif //STUDENT_MANAGEMENT_SYSTEM_MANAGEMENT_SYSTEM_TYPES_H

```