PROGRAMING PRACTICES
ASSIGNMENT (MINI_PROJECT)
DEBUGGING (STUDENT MANAGEMENT SYSTEM)

```
/*
STUDENT MANAGEMENT SYSTEM
CREATED BY : MUSTAFA KASUBHAI WALA
ROLL NO. 0801CS221093.
© COPYRIGHT 2023 BY MUSTAFA KASUBHAI WALA . THESE MATERIAL CANNOT BE
DUPLICATE FOR ANY PROFIT DRIVEN ENTERPRISE.
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define g_size 60
#define g_N 100

struct Grades {
    char c_subjects[g_size];
    char c_grade[g_size];
};

struct studentinfo {
    int i_year_admission;
    char c_enrollment_no[g_size];
    char c_name[g_size];
    char c_lastname[g_size];
    int i_phone_no;
    int i_age;
    char c_City[g_size];
    char c_Gender[g_size];
    char c_address[g_size];
    char c_email_id[g_size];
    char c_branch[g_size];
    int i_fees;
    struct Grades gradesshow[g_N];
    int i_num_subjects;
};
void F_Add_student_details(struct studentinfo *student, int
*serial_no);
```

```c
void F_Display_student_details(struct studentinfo *student, int
*serial_no);
int F_Search_the_student(struct studentinfo *student, int *serial_no,
char *c_enrollment);
void F_Delete_the_student_details(struct studentinfo *student, int
*serial_no, char *c_enrollment);
void F_Modify_the_student_details(struct studentinfo *student, int
*serial_no, char *c_enrollment);
void F_Student_grades(struct studentinfo *student, int *serial_no, char
*c_enrollment);
void F_Student_grades_show(struct studentinfo *student, int *serial_no,
char *c_enrollment);

int main() {
    struct studentinfo *student = malloc(g_N * sizeof(struct
studentinfo));

    if (student == NULL) {
        fprintf(stderr, "Memory allocation failed\n");
        return 1;
    }
    int serial_no = 0;
    int choice;
    char c_enrollment_no[g_size];
     while (1) {
        printf("Welcome to Student Management System:\n");
        printf("1. Add Student\n");
        printf("2. Display Student Details\n");
        printf("3. Search Student\n");
        printf("4. Delete Student Registration\n");
        printf("5. Modify Student Details\n");
        printf("6. Add Grades\n");
        printf("7. Display Grades\n");
        printf("8. Exit\n");
        printf("Enter your number ");
        scanf("%d", choice);

        switch (choice) {
            case 1:
            // CALL THE FUNCTION WHICH TAKES DETAILS OF THE STUDENT .
                F_Add_student_details(&student[serial_no], &serial_no);
                break;
            case 2:
```

```
            // CALL THE FUNCTION WHICH DISPLAY THE DETAILS OF THE
STUDENT .
                F_Display_student_details(student, &serial_no);
                break;
            case 3:
            // CALL THE FUNCTION WHICH SEARCH THE STUDENT EXISTS OR NOT
.
                printf("Enter the student Enrollment number for
search:\n");
                scanf("%s", c_enrollment_no);
                int index = F_Search_the_student(student, &serial_no,
c_enrollment_no);
                if (index != -1) {

printf("*******----------------------------------------*******\n");
                    printf("Enrollment No:%s\n",
student[index].c_enrollment_no);
                    printf("Year of admission: %d\n",
student[index].i_year_admission);
                    printf("Name: %s %s\n", student[index].c_name,
student[index].c_lastname);
                    printf("Gender:%s",student[index].c_Gender);
                    printf("Branch: %s\n", student[index].c_branch);
                    printf("Address :%s\n",student[index].c_address);
                    printf("city: %s",student[index].c_City);
                    printf("Age of the student: %d\n",
student[index].i_age);
                    printf("Student phone no is: %d\n",
student[index].i_phone_no);
                    printf("Email
Id:%s\n",student[index].c_email_id);
                    printf("****----------------------****\n");
                } else {
                    printf("Student details not found.\n");
                }
                break;
            case 4:
            // CALL THE FUNCTION TO DELETE THE STUDENT DETAILS AND FREE
THE SPACE.
                printf("Enter the student Enrollment number whose
registration is cancelled: ");
                scanf("%s", c_enrollment_no);
```

```
                F_Delete_the_student_details(student, &serial_no,
c_enrollment_no);
                break;
            case 5:
            // CALL THE FUNCTION TO MODIFY THE DETAILS OF THE STUDENT
                printf("Enter the Enrollment Number of the student to
modify details:\n");
                scanf("%s", c_enrollment_no);
                F_Modify_the_student_details(student, &serial_no,
c_enrollment_no);
                break;

            case 6:
            // CALL THE FUNCTION FOR GRADING
               printf("Enter the Enrollment number of student for
grades:\n");
                scanf("%s", c_enrollment_no);
                F_Student_grades(student, &serial_no, c_enrollment_no);
                break;
            case 7:
            // CALL THE FUNCTION TO DISPLAY THE GRADES OF THE STUDENT
                printf("Enter the Enrollment number of student to
display grades:\n");
                scanf("%s", c_enrollment_no);
                F_Student_grades_show(student, &serial_no,
c_enrollment_no);
                break;
            case 8:
            // IN THESE CASE WHILE TERMINATE .
                free(student);
                printf("Thank you for using Student Management
System.\n");
                return 0;
            default:
                printf("Invalid choice. Please try again.\n");
        }
    }
}
void F_Add_student_details(struct studentinfo *student, int *serial_no)
{
    printf("Enter the data for student %d:\n", (*serial_no) + 1);
    printf("*********************************\n");
    printf("Enter the year of admission: ");
```

```
    scanf("%d", &student->i_year_admission);
    printf("Enter the Enrollment no of student: ");
    scanf("%s", student->c_enrollment_no);
    printf("Enter the student name: ");
    scanf("%s", student->c_name);
    printf("Enter the surname: ");
    scanf("%s", student->c_lastname);
    printf("Enter the Gender :");
    scanf("%s",student->c_Gender);
    printf("Enter the branch: ");
    scanf("%s", student->c_branch);
    printf("Enter the phone number: ");
    scanf("%d",&student->i_phone_no);
    printf("Enter the address:");
    scanf("%s",student->c_address);
    printf("Enter the city :");
    scanf("%s",student->c_City );
    printf("Enter the age of the student:");
    scanf("%d", &student->i_age);
    printf("Enter the Email Id: ");
    scanf("%s",student->c_email_id);
    printf("Enter the Fees: ");
    scanf("%d", &student->i_fees);
    printf("Detail of Student is Saved :\n");
    printf("\n*****************************\n");

    (*serial_no)++;
}
void F_Display_student_details(struct studentinfo *student, int
*serial_no) {
   printf("Details of students are as follows:\n");
    for (int i = 0; i < *serial_no; i++)
    {
        printf("****----------------------------------******\n");
        printf("Enrollment No:%s\n", student[i].c_enrollment_no);
        printf("Year of admission: %d\n", student[i].i_year_admission);
        printf("Name: %s %s\n", student[i].c_name,
student[i].c_lastname);
        printf("Gender:%s",student[i]. c_Gender);
        printf("Branch: %s\n", student[i].c_branch);
        printf("Address :%s\n",student[i].c_address);
        printf("city: %s",student[i].c_City);
        printf("Age of the student: %d\n", student[i].i_age);
```

```
        printf("Student phone no is: %d\n", student[i].i_phone_no);
        printf("Email Id:%s\n",student[i].c_email_id);
        printf("Fees: %d\n", student[i].i_fees);
        printf("****------------------------****\n");
    }
}
int F_Search_the_student(struct studentinfo *student, int *serial_no,
char *enrollment) {
       for (int i = 0; i < *serial_no; i++)
    {
        if (strcmp(student[i].c_enrollment_no, enrollment) == 0)
        {
            return i;
        }
    }
    return -1;
}
void F_Delete_the_student_details(struct studentinfo *student, int
*serial_no, char *c_enrollment) {
    int search = F_Search_the_student(student, serial_no,
c_enrollment);
    if (search != -1)
    {
        for (int i = search; i < *serial_no - 1; i++)
        {
            student[i] = student[i + 1];
        }
        (*serial_no)--;
        printf("Student with enrollment number: %s has been
deleted.\n", c_enrollment);
    }
    else
    {
        printf("Student with enrollment number: %s is not found.\n",
c_enrollment);
    }
}
void F_Modify_the_student_details(struct studentinfo *student, int
*serial_no, char *c_enrollment) {
   int index = F_Search_the_student(student, serial_no, c_enrollment);
    if (index != -1)
    {
```

```
printf("-------------------------------------------------------------");
        printf("Enter the modified details for student with Enrollment
Number %s:\n",c_enrollment);
        printf("Enter the new Year of Admission: ");
        scanf("%d", &student[index].i_year_admission);
        printf("Enter the new Name and Last Name of the student: ");
        scanf("%s %s", student[index].c_name,
student[index].c_lastname);
        printf("Enter the new Phone Number: ");
        scanf("%d", &student[index].i_phone_no);
        printf("Enter the new Age of Student: ");
        scanf("%d", &student[index].i_age);
        printf("Enter the new Email Address: ");
        scanf("%s", student[index].c_email_id);
        printf("Enter the new City: ");
        scanf("%s", student[index].c_City);
        printf("Student details updated successfully.\n");
        printf("-----------------------------------------\n");
    }
    else
    {
        printf("Student with Enrollment Number %s not found.\n",
c_enrollment);
    }
}
void F_Student_grades(struct studentinfo *student, int *serial_no, char
*c_enrollment) {
    int index = F_Search_the_student(student, serial_no, c_enrollment);
    if (index != -1) {
        printf("Enter the number of subjects:\n");
        scanf("%d", &student[index].i_num_subjects);
        for (int i = 0; i < student[index].i_num_subjects; i++) {
            printf("Enter subject %d: ", i + 1);
            scanf("%s", student[index].gradesshow[i].c_subjects);
            printf("Enter the grade for %s: ",
student[index].gradesshow[i].c_subjects);
            scanf("%s", student[index].gradesshow[i].c_grade);
        }
        printf("Grades for student with Enrollment No. %s successfully
added.\n", c_enrollment);
    } else {
```

```
        printf("Student with Enrollment %s not found.\n",
c_enrollment);
    }
}
void F_Student_grades_show(struct studentinfo *student, int *serial_no,
char *c_enrollment) {
    int index =F_Search_the_student(student, serial_no, c_enrollment);
    if (index != -1) {
        printf("\n--------------------------\n");
        printf("Grades for student with Enrollment No. %s:\n",
student[index].c_enrollment_no);
        for (int i = 0; i < student[index].i_num_subjects; i++) {
            printf("Subject: %s, Grade: %s\n",
student[index].gradesshow[i].c_subjects,
student[index].gradesshow[i].c_grade);
        printf ("\n--------------------------\n");
        }
    } else {
        printf("Enrollment %s not found.\n", c_enrollment);
    }
}
```

```
main.c:68:17: warning: format '%d' expects argument of type 'int *', but argument 2 has type
 'int' [-Wformat=]
Reading symbols from a.out...
(gdb) list
32          char c_branch[g_size];
33          int i_fees;
34          struct Grades gradesshow[g_N];
35          int i_num_subjects;
36      };
37      void F_Add_student_details(struct studentinfo *student, int *serial_no);
38      void F_Display_student_details(struct studentinfo *student, int *serial_no);
39      int F_Search_the_student(struct studentinfo *student, int *serial_no, char *c_enroll
ment);
40      void F_Delete_the_student_details(struct studentinfo *student, int *serial_no, char
*c_enrollment);
41      void F_Modify_the_student_details(struct studentinfo *student, int *serial_no, char
*c_enrollment);
(gdb) break main
Breakpoint 1 at 0x1255: file main.c, line 46.
(gdb) r
Starting program: /home/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
```

```
Breakpoint 1, main () at main.c:46
46      int main() {
(gdb) n
47          struct studentinfo *student = malloc(g_N * sizeof(struct studentinfo));
(gdb) n
50          if (student == NULL) {
(gdb) n
54          int serial_no = 0;
(gdb) n
58              printf("Welcome to Student Management System:\n");
(gdb) n
Welcome to Student Management System:
59              printf("1. Add Student\n");
(gdb) n
1. Add Student
60              printf("2. Display Student Details\n");
(gdb) n
2. Display Student Details
61              printf("3. Search Student\n");
(gdb) n
```

```
2. Display Student Details
61              printf("3. Search Student\n");
(gdb) n
3. Search Student
62              printf("4. Delete Student Registration\n");
(gdb) n
4. Delete Student Registration
63              printf("5. Modify Student Details\n");
(gdb) n
5. Modify Student Details
64              printf("6. Add Grades\n");
(gdb) n
6. Add Grades
65              printf("7. Display Grades\n");
(gdb) n
7. Display Grades
66              printf("8. Exit\n");
(gdb) n
8. Exit
67              printf("Enter your number ");
(gdb) n
68              scanf("%d", choice);
(gdb) n
Enter your number 1
```

```
(gdb) n
7. Display Grades
66              printf("8. Exit\n");
(gdb) n
8. Exit
67              printf("Enter your number ");
(gdb) n
68              scanf("%d", choice);
(gdb) n
Enter your number 1

Program received signal SIGSEGV, Segmentation fault.
0x00007ffff7def1c9 in __vfscanf_internal (s=<optimized out>, format=<optimized out>, argp
argptr@entry=0x7fffffffea70, mode_flags=mode_flags@entry=2) at ./stdio-common/vfscanf-int
al.c:1896
1896    ./stdio-common/vfscanf-internal.c: No such file or directory.
(gdb) n

Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.
(gdb) n
The program is not being run.
(gdb) n
```

AFTER DEBUGGING THE CODE .

```
Reading symbols from a.out...
(gdb) break main
Note: breakpoint 1 also set at pc 0x1255.
Breakpoint 2 at 0x1255: file main.c, line 46.
(gdb) r
Starting program: /home/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, main () at main.c:46
46      int main() {
(gdb) n
47          struct studentinfo *student = malloc(g_N * sizeof(struct studentinfo));
(gdb) n
50          if (student == NULL) {
(gdb) n
54          int serial_no = 0;
(gdb) n
58              printf("Welcome to Student Management System:\n");
(gdb) n
Welcome to Student Management System:
59              printf("1. Add Student\n");
(gdb) n
1. Add Student
```

```
Welcome to Student Management System:
59              printf("1. Add Student\n");
(gdb) n
1. Add Student
60              printf("2. Display Student Details\n");
(gdb) n
2. Display Student Details
61              printf("3. Search Student\n");
(gdb) n
3. Search Student
62              printf("4. Delete Student Registration\n");
(gdb) n
4. Delete Student Registration
63              printf("5. Modify Student Details\n");
(gdb) n
5. Modify Student Details
64              printf("6. Add Grades\n");
(gdb) n
6. Add Grades
65              printf("7. Display Grades\n");
(gdb) n
7. Display Grades
66              printf("8. Exit\n");
(gdb) n
```

```
(gdb) n
Enter your number 1
71              switch (choice) {
(gdb) n
74                      F_Add_student_details(&student[serial_no], &serial_no);
(gdb) n
Enter the data for student 1:
**********************************
Enter the year of admission: 2015
Enter the Enrollment no of student: 0801cs221093
Enter the student name: mustafa
Enter the surname: kasu
Enter the Gender :male
Enter the branch: cse
Enter the phone number: 7894561589
Enter the address:sgsits
Enter the city :indore
Enter the age of the student:18
Enter the Email Id: mustafa@gamil.com
Enter the Fees: 45000
Detail of Student is Saved :

*****************************
75                      break;
```

```
Welcome to Student Management System:
59              printf("1. Add Student\n");
(gdb) n
1. Add Student
60              printf("2. Display Student Details\n");
(gdb) n
2. Display Student Details
61              printf("3. Search Student\n");
(gdb) n
3. Search Student
62              printf("4. Delete Student Registration\n");
(gdb) n
4. Delete Student Registration
63              printf("5. Modify Student Details\n");
(gdb) n
5. Modify Student Details
64              printf("6. Add Grades\n");
(gdb) n
6. Add Grades
65              printf("7. Display Grades\n");
(gdb) n
7. Display Grades
66              printf("8. Exit\n");
(gdb) n
```

```
68              scanf("%d", &choice);
(gdb) n
Enter your number 2
71              switch (choice) {
(gdb) n
78                      F_Display_student_details(student, &serial_no);
(gdb) n
Details of students are as follows:
****----------------------------------******
Enrollment No:0801cs221093
Year of admission: 2015
Name: mustafa kasu
Gender:maleBranch: cse
Address :sgsits
city: indoreAge of the student: 18
Student phone no is: -695373003
Email Id:mustafa@gamil.com
Fees: 45000
****-----------------------****
79                      break;
(gdb) n
58              printf("Welcome to Student Management System:\n");
(gdb) n
Welcome to Student Management System:
```

```
8. Exit
67              printf("Enter your number ");
(gdb) n
68              scanf("%d", &choice);
(gdb) n
Enter your number 8
71              switch (choice) {
(gdb) n
129                     free(student);
(gdb) n
130                     printf("Thank you for using Student Management System.\n");
(gdb) n
Thank you for using Student Management System.
131                     return 0;
(gdb) n
136     }
(gdb) n
__libc_start_call_main (main=main@entry=0x555555555249 <main>, argc=argc@entry=1, argv=argv@e
ry=0x7fffffffecd8) at ../sysdeps/nptl/libc_start_call_main.h:74
74      ../sysdeps/nptl/libc_start_call_main.h: No such file or directory.
(gdb) n
[Inferior 1 (process 875) exited normally]
(gdb) n
The program is not being run.
```