Institute of Computer Technology

B. Tech. Computer Science and Engineering

Sub: ESFP – II

Course Code: 2CSE203

Practical – 2

Name: Jaymin Gondaliya

Enrollment No: 23162171007

Sem – 2

Branch: CS

Class: B

Batch: 25

Objective:

To learn DMA (Dynamic memory allocation)

Problem Definition-1:

Vivek engineering college, which is situated at Pune IT park. College authority decides to come up with a new idea for the handling of examination seating arrangements as per the rules provided by the university for different-different courses. For that, the college examination committee department wants separate records, for those students, who are giving a remedial or regular examination for the semester-II, IV & VI in the given current academic calendar month. For that, the examination committee wants to take all the basic information related to students like rollno, name, class, semester, subject, and exam fee. So, whenever is required to search any student records by id, or by name, he can search randomly, if the committee found some rectification is required in the student record, he can modify / update the given record by id or by name, if by mistakenly student filled up the examination form for the given said semester, than committee should have the authority to delete the student record by id or by their name from the exam record.

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```c
struct Student
{
    int rollno;
    char name[50];
    char std[10];
    char sem[5];
    char subject[30];
    int fee;
};

void display(struct Student *students, int n);
void modify(struct Student **students, int *n);
void delete(struct Student **students, int *n);
void search(struct Student *students, int n);

int main()
{
    int n;

    printf("Enter how many records you want to store: ");
    scanf("%d", &n);

    struct Student *students = (struct Student *)malloc(n * sizeof(struct Student));

    printf("Enter no, name, std, sem, subject and fee:\n");
    for (int i = 0; i < n; i++)
    {
        scanf("%d %s %s %s %s %d", &students[i].rollno, students[i].name, students[i].std, students[i].sem, students[i].subject, &students[i].fee);
    }

    printf("========== Output of student information===========\n");
    display(students, n);

    char modifyChoice;
    printf("Do you want to modify/update records (Y/N)? ");
    scanf(" %c", &modifyChoice);
    if (modifyChoice == 'Y' || modifyChoice == 'y')
    {
        modify(&students, &n);
    }

    char deleteChoice;
    printf("Do you want to delete (Y/N)? ");
    scanf(" %c", &deleteChoice);
    if (deleteChoice == 'Y' || deleteChoice == 'y')
    {
```

```c
        delete (&students, &n);
    }

    search(students, n);

    free(students);

    return 0;
}

void display(struct Student *students, int n)
{
    for (int i = 0; i < n; i++)
    {
        printf("%d %s %s %s %s %d\n", students[i].rollno, students[i].name,
students[i].std, students[i].sem, students[i].subject, students[i].fee);
    }
}

void modify(struct Student **students, int *n)
{
    int choice;
    printf("How do you want to modify record by id or by name?\nPress <1> for
by id and Press <2> for by name? ");
    scanf("%d", &choice);

    if (choice == 1)
    {
        int modifyID;
        printf("Enter student id: ");
        scanf("%d", &modifyID);

        *students = (struct Student *)realloc(*students, (*n + 1) *
sizeof(struct Student));

        for (int i = 0; i < *n; i++)
        {
            if ((*students)[i].rollno == modifyID)
            {
                printf("%d %s %s %s %s %d\n", (*students)[i].rollno,
(*students)[i].name, (*students)[i].std, (*students)[i].sem,
(*students)[i].subject, (*students)[i].fee);
                printf("Enter no, name, std, sem, subject and fee:\n");
                scanf("%d %s %s %s %s %d", &(*students)[i].rollno,
(*students)[i].name, (*students)[i].std, (*students)[i].sem,
(*students)[i].subject, &(*students)[i].fee);
                printf("Record Updated successfully\n");
                break;
```

```c
            }
        }
        (*n)++;
    }
    else if (choice == 2)
    {
        char modifyName[50];
        printf("Enter student name: ");
        scanf("%s", modifyName);

        *students = (struct Student *)realloc(*students, (*n + 1) *
sizeof(struct Student));

        for (int i = 0; i < *n; i++)
        {
            if (strcmp((*students)[i].name, modifyName) == 0)
            {
                printf("%d %s %s %s %s %d\n", (*students)[i].rollno,
(*students)[i].name, (*students)[i].std, (*students)[i].sem,
(*students)[i].subject, (*students)[i].fee);
                printf("Enter no, name, std, sem, subject and fee:\n");
                scanf("%d %s %s %s %s %d", &(*students)[i].rollno,
(*students)[i].name, (*students)[i].std, (*students)[i].sem,
(*students)[i].subject, &(*students)[i].fee);
                printf("Record Updated successfully\n");
                break;
            }
        }
        (*n)++;
    }
}

void delete(struct Student **students, int *n)
{
    int choice;
    printf("How do you want to delete record by id or by name?\nPress <1> for
by id and Press <2> for by name? ");
    scanf("%d", &choice);

    if (choice == 1)
    {
        int deleteID;
        printf("Enter student id: ");
        scanf("%d", &deleteID);

        for (int i = 0; i < *n; i++)
        {
            if ((*students)[i].rollno == deleteID)
```

```c
        {
            printf("%d %s %s %s %s %d\n", (*students)[i].rollno,
(*students)[i].name, (*students)[i].std, (*students)[i].sem,
(*students)[i].subject, (*students)[i].fee);
            for (int j = i; j < (*n - 1); j++)
            {
                (*students)[j] = (*students)[j + 1];
            }
            *students = (struct Student *)realloc(*students, (*n - 1) *
sizeof(struct Student));
            (*n)--;
            printf("Record deleted successfully\n");
            break;
        }
    }
}
else if (choice == 2)
{
    char deleteName[50];
    printf("Enter student name: ");
    scanf("%s", deleteName);

    for (int i = 0; i < *n; i++)
    {
        if (strcmp((*students)[i].name, deleteName) == 0)
        {
            printf("%d %s %s %s %s %d\n", (*students)[i].rollno,
(*students)[i].name, (*students)[i].std, (*students)[i].sem,
(*students)[i].subject, (*students)[i].fee);
            for (int j = i; j < (*n - 1); j++)
            {
                (*students)[j] = (*students)[j + 1];
            }
            *students = (struct Student *)realloc(*students, (*n - 1) *
sizeof(struct Student));
            (*n)--;
            printf("Record deleted successfully\n");
            break;
        }
    }
}
}

void search(struct Student *students, int n)
{
    char searchName[50];
    printf("Find the student record by name:\nEnter student name: ");
    scanf("%s", searchName);
```

```
    int found = 0;
    for (int i = 0; i < n; i++)
    {
        if (strcmp(students[i].name, searchName) == 0)
        {
            printf("%d %s %s %s %s %d\n", students[i].rollno,
students[i].name, students[i].std, students[i].sem, students[i].subject,
students[i].fee);
            found = 1;
            break;
        }
    }

    if (!found)
    {
        printf("Student with name '%s' not found.\n", searchName);
    }
}
```

Output-

```
C:\ICT\SEM-2\ESFP-II\main.exe
Enter how many records you want to store:3
 Enter no, name, std, sem, subject and fee:
64 test first II esfp 2500
86 dev third I CO 6500
33 root ten III IP 5000
========== Output of student information===========
64 test first II esfp 2500
86 dev third I CO 6500
33 root ten III IP 5000
Do you want to modify/update records (Y/N)?y
 How do you want to modify record by id or by name?
Press <1> for by id and Press <2> for by name?2
 Enter student name:root
 33 root ten III IP 5000
Enter no, name, std, sem, subject and fee:
33 toor ten III IP 5500
Record Updated successfully
Do you want to delete (Y/N)?y
 How do you want to delete record by id or by name?
Press <1> for by id and Press <2> for by name?1
 Enter student id:64
 64 test first II esfp 2500
Record deleted successfully
Find the student record by name:
Enter student name:dev
 86 dev third I CO 6500


Process finished with exit code 0
```