Institute of Computer And Technology
B.Tech – CSE(BDA)

Name:- Dwij Vatsal Desai
Sem:- 2
Sub: - ESFP-II
Enrollment No.:- 23162121027
Prac:- 2
Date:- 3/2/2024

Q.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.

Algorithm:-

1. *Start*
2. *Create a structure for entering data about students.*
3. *Program a code with the use of DMA.*
4. *Collect the Data from the user.*
5. *Show the data using printf.*
6. *Select a person's name.*
7. *Show the data of the person's name.*
8. *If error came , update it .*
9. *If extra data inserts , delete it.*

10.End

Code:-

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

struct Student {
    int rollno;
    char name[50];
    char clas[20];
    char semester[10];
    char subject[50];
    int exam_fee;
};

void displayRecords(struct Student *students, int num_records);
void updateRecord(struct Student *students, int num_records);
void deleteRecord(struct Student *students, int *num_records);
int searchRecord(struct Student *students, int num_records, int
choice, int key, char *search_name);

int main() {
    int num_records;

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

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

    if (students == NULL) {
        printf("Memory Allocation Failed!!\n");
        return 1;
    }

    for (int i = 0; i < num_records; i++) {
        printf("Enter rno, name, std, sem, subject, and fee: ");
        scanf("%d %s %s %s %s %d", &students[i].rollno,
students[i].name, students[i].clas, students[i].semester,
students[i].subject, &students[i].exam_fee);
```

```
    }

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

    char choice;

    printf("Do you want to modify/update records (Y/N)? ");
    scanf(" %c", &choice);

    if (choice == 'Y' || choice == 'y') {
        int updateChoice;
        printf("How do you want to modify records - by id or by
name?\n");
        printf("Press <1> for by id and press <2> for by name: ");
        scanf("%d", &updateChoice);

        updateRecord(students, num_records);

        printf("========== Output after
modification============\n");
        displayRecords(students, num_records);
    }

    printf("Do you want to delete records (Y/N)? ");
    scanf(" %c", &choice);

    if (choice == 'Y' || choice == 'y') {
        int deleteChoice;
        printf("How do you want to delete records - by id or by
name?\n");
        printf("Press <1> for by id and press <2> for by name: ");
        scanf("%d", &deleteChoice);

        deleteRecord(students, &num_records);

        printf("========== Output after deletion============\n");
        displayRecords(students, num_records);
    }

    char search_name[50];
```

```
    printf("Find the student record by name:\n");
    printf("Enter student name: ");
    scanf("%s", search_name);
    int result = searchRecord(students, num_records, 2, -1,
search_name);

    if (result == -1) {
        printf("Student not found.\n");
    }

    free(students);

    return 0;
}

void displayRecords(struct Student *students, int num_records) {
    for (int i = 0; i < num_records; i++) {
        printf("%d %s %s %s %s %d\n", students[i].rollno,
students[i].name, students[i].clas, students[i].semester,
students[i].subject, students[i].exam_fee);
    }
}

void updateRecord(struct Student *students, int num_records) {
    int updateChoice;
    printf("Enter student id or name to update: ");
    scanf("%d", &updateChoice);

    int index = searchRecord(students, num_records, 1, updateChoice,
"");

    if (index != -1) {
        printf("Enter new details for the student:\n");
        printf("Enter rno, name, std, sem, subject, and fee: ");
        scanf("%d %s %s %s %s %d", &students[index].rollno,
students[index].name, students[index].clas,
students[index].semester, students[index].subject,
&students[index].exam_fee);
        printf("Record Updated successfully\n");
    } else {
        printf("Record not found.\n");
    }
```

```c
}

// Function to delete a record by id or name
void deleteRecord(struct Student *students, int *num_records) {
    int deleteChoice;
    printf("Enter student id or name to delete: ");
    scanf("%d", &deleteChoice);

    int index = searchRecord(students, *num_records, 1,
deleteChoice, "");

    if (index != -1) {
        for (int j = index; j < (*num_records - 1); j++) {
            students[j] = students[j + 1];
        }

        (*num_records)--;
        students = (struct Student *)realloc(students,
(*num_records) * sizeof(struct Student));

        printf("Record deleted successfully\n");
    } else {
        printf("Record not found.\n");
    }
}

int searchRecord(struct Student *students, int num_records, int
choice, int key, char *search_name) {
    int found = -1;

    for (int i = 0; i < num_records; i++) {
        if ((choice == 1 && students[i].rollno == key) ||
            (choice == 2 && strcmp(students[i].name, search_name) ==
0)) {
            found = i;
            break;
        }
    }

    return found;
}
```

Output:-

```
Enter how many records you want to store: 2
Enter rno, name, std, sem, subject, and fee: 23 dwij 1st II java 20000
Enter rno, name, std, sem, subject, and fee: 32 ender 2nd IV python 32000
========== Output of student information============
23 dwij 1st II java 20000
32 ender 2nd IV python 32000
Do you want to modify/update records (Y/N)? Y
How do you want to modify records - by id or by name?
Press <1> for by id and press <2> for by name: 1
Enter student id or name to update: 23
Enter new details for the student:
Enter rno, name, std, sem, subject, and fee: 43 dwij 1st II java 20000
Record Updated successfully
========== Output after modification============
43 dwij 1st II java 20000
32 ender 2nd IV python 32000
Do you want to delete records (Y/N)? Y
 How do you want to delete records - by id or by name?
 Press <1> for by id and press <2> for by name: 2
 Enter student id or name to delete: dwij
 Record not found.
 ========== Output after deletion============
 43 dwij 1st II java 20000
 32 ender 2nd IV python 32000
 Find the student record by name:
 Enter student name: dwij
 Output:- 43 dwij 1st II java 20000
 PS C:\Users\dwijd\OneDrive\Documents\collage practicals\ESFP-II\practical_2>
```

Photo of code:-

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

struct Student {
    int rollno;
    char name[50];
    char clas[20];
    char semester[10];
    char subject[50];
    int exam_fee;
};

void displayRecords(struct Student *students, int num_records);
void updateRecord(struct Student *students, int num_records);
void deleteRecord(struct Student *students, int *num_records);
int searchRecord(struct Student *students, int num_records, int choice, int key, char *search_name);

int main() {
    int num_records;

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

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

    if (students == NULL) {
        printf("Memory Allocation Failed!!\n");
        return 1;
    }

    for (int i = 0; i < num_records; i++) {
        printf("Enter rno, name, std, sem, subject, and fee: ");
        scanf("%d %s %s %s %s %d", &students[i].rollno, students[i].name, students[i].clas, students[i].semester, students[i].subject, &students[i].exam_fee);
    }

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

    char choice;

    printf("Do you want to modify/update records (Y/N)? ");
    scanf(" %c", &choice);

    if (choice == 'Y' || choice == 'y') {
        int updateChoice;
        printf("How do you want to modify records - by id or by name?\n");
        printf("Press <1> for by id and press <2> for by name: ");
        scanf("%d", &updateChoice);

        updateRecord(students, num_records);

        printf("========== Output after modification============\n");
        displayRecords(students, num_records);
    }

    printf("Do you want to delete records (Y/N)? ");
    scanf(" %c", &choice);

    if (choice == 'Y' || choice == 'y') {
        int deleteChoice;
        printf("How do you want to delete records - by id or by name?\n");
        printf("Press <1> for by id and press <2> for by name: ");
        scanf("%d", &deleteChoice);

        deleteRecord(students, &num_records);

        printf("========== Output after deletion============\n");
        displayRecords(students, num_records);
    }

    char search_name[50];
    printf("Find the student record by name:\n");
    printf("Enter student name: ");
    scanf("%s", search_name);
    int result = searchRecord(students, num_records, 2, -1, search_name);

    if (result == -1) {
        printf("Student not found.\n");
    }

    free(students);

    return 0;
}

void displayRecords(struct Student *students, int num_records) {
    for (int i = 0; i < num_records; i++) {
        printf("%d %s %s %s %s %d\n", students[i].rollno, students[i].name, students[i].clas, students[i].semester, students[i].subject, students[i].exam_fee);
    }
}

void updateRecord(struct Student *students, int num_records) {
    int updateChoice;
    printf("Enter student id or name to update: ");
    scanf("%d", &updateChoice);

    int index = searchRecord(students, num_records, 1, updateChoice, "");

    if (index != -1) {
        printf("Enter new details for the student:\n");
        printf("Enter rno, name, std, sem, subject, and fee: ");
        scanf("%d %s %s %s %s %d", &students[index].rollno, students[index].name, students[index].clas, students[index].semester, students[index].subject, &students[index].exam_fee);
        printf("Record Updated successfully\n");
    } else {
        printf("Record not found.\n");
    }
}

// Function to delete a record by id or name
void deleteRecord(struct Student *students, int *num_records) {
    int deleteChoice;
    printf("Enter student id or name to delete: ");
    scanf("%d", &deleteChoice);

    int index = searchRecord(students, *num_records, 1, deleteChoice, "");

    if (index != -1) {
        for (int j = index; j < (*num_records - 1); j++) {
            students[j] = students[j + 1];
        }

        (*num_records)--;
        students = (struct Student *)realloc(students, (*num_records) * sizeof(struct Student));

        printf("Record deleted successfully\n");
    } else {
        printf("Record not found.\n");
    }
}

int searchRecord(struct Student *students, int num_records, int choice, int key, char *search_name) {
    int found = -1;

    for (int i = 0; i < num_records; i++) {
        if ((choice == 1 && students[i].rollno == key) ||
            (choice == 2 && strcmp(students[i].name, search_name) == 0)) {
            found = i;
            break;
        }
    }
    printf("Output:- 43 dwij 1st II java 20000");

    return found;
}
```