| | | |
|---|---|---|
| **S.No: 1** | Exp. Name: ***Project Module*** | **Date: 2024-06-14** |

**Aim:**

Project Module

**Source Code:**

```
hello.c
```

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

#define MAX_BATCHES 10
#define MAX_STUDENTS 50

typedef struct {
    char name[50];
    int rollNo;
    float marks;
} Student;

Student students[MAX_BATCHES][MAX_STUDENTS];
int studentCount[MAX_BATCHES];

int findStudent(int batch, char* name) {
    for (int i = 0; i < studentCount[batch]; i++) {
        if (strcmp(students[batch][i].name, name) == 0) {
            return i;
        }
    }
    return -1;
}

void addStudent(int batch, char* name, int rollNo, float marks) {
    if (studentCount[batch] < MAX_STUDENTS) {
        for (int i = 0; i < studentCount[batch]; i++) {
            if (strcmp(students[batch][i].name, name) == 0) {
                printf("Student already exists in this batch.\n");
                return;
            }
        }
        students[batch][studentCount[batch]].name[5] = '\0';
        strcpy(students[batch][studentCount[batch]].name, name);
        students[batch][studentCount[batch]].rollNo = rollNo;
        students[batch][studentCount[batch]].marks = marks;
        studentCount[batch]++;
    } else {
        printf("Maximum number of students reached for this batch.\n");
    }
}

void deleteStudent(int batch, char* name) {
    int index = findStudent(batch, name);
    if (index != -1) {
        for (int i = index; i < studentCount[batch] - 1; i++) {
            students[batch][i] = students[batch][i + 1];
        }
        studentCount[batch]--;
    } else {
        printf("Student not found in this batch.\n");
    }
}
```

```
    if (index != -1) {
        printf("Student found in this batch:\n");
        printf("Name: %s\n", students[batch][index].name);
        printf("Roll number: %d\n", students[batch][index].rollNo);
        printf("Marks: %f\n", students[batch][index].marks);
    } else {
        printf("Student not found in this batch.\n");
    }
}

void printStudents(int batch) {
    if (studentCount[batch] > 0) {
        printf("Batch %d:\n", batch);
        for (int i = 0; i < studentCount[batch]; i++) {
            printf("Student %d:\n", i + 1);
            printf("Name: %s\n", students[batch][i].name);
            printf("Roll number: %d\n", students[batch][i].rollNo);
            printf("Marks: %f\n", students[batch][i].marks);
            printf("\n");
        }
    } else {
        printf("No students found in this batch.\n");
    }
}

int main() {
    int batch;
    char choice;
    char name[50];

    // Initialize student count for each batch
    for (int i = 0; i < MAX_BATCHES; i++) {
        studentCount[i] = 0;
    }

    while (1) {
        printf("Enter choice:\n");
        printf("A - Add student\n");
        printf("D - Delete student\n");
        printf("S - Search student\n");
        printf("P - Print students\n");
        printf("Q - Quit\n");
        scanf(" %c", &choice);

        switch (choice) {
            case 'A':
            case 'a':
                printf("Enter batch number: ");
                scanf("%d", &batch);
                if (batch < 0 || batch >= MAX_BATCHES) {
                    printf("Invalid batch number. Please enter a valid batch 
number.\n");
                } else {
                    printf("Enter name: ");
                    scanf("%s", name);
                    printf("Enter marks: ");
                    scanf("%f", &students[batch][studentCount[batch]].marks);
                    addStudent(batch, name, students[batch]
[studentCount[batch]].rollNo, students[batch][studentCount[batch]].marks);
                }
                break;
            case 'D':
            case 'd':
                printf("Enter batch number: ");
                scanf("%d", &batch);
                if (batch < 0 || batch >= MAX_BATCHES) {
                    printf("Invalid batch number. Please enter a valid batch
number.\n");
                } else {
                    printf("Enter name: ");
                    scanf("%s", name);
                    deleteStudent(batch, name);
                }
                break;
            case 'S':
            case 's':
                printf("Enter batch number: ");
                scanf("%d", &batch);
                if (batch < 0 || batch >= MAX_BATCHES) {
                    printf("Invalid batch number. Please enter a valid batch
number.\n");
                } else {
                    printf("Enter name: ");
                    scanf("%s", name);
                    searchStudent(batch, name);
                }
                break;
            case 'P':
            case 'p':
                printf("Enter batch number: ");
                scanf("%d", &batch);
                printStudents(batch);
                break;
            case 'Q':
            case 'q':
                return 0;
            default:
                printf("Invalid choice. Please enter a valid choice.\n");
        }
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |

```
Hello World
```