CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

Bachelor of computer application (BCA)

Project

Data Structure

Submitted By:

Name: Piyush jana

UID: 24bca10472

Class & section: 24bca7(A)

Supervision By:

Name: Monika Choudhary

Designation: Assistant Professor

Student Record system

AIM: To Develop Student Record System Using Data Structure in C

Procedure Code: -

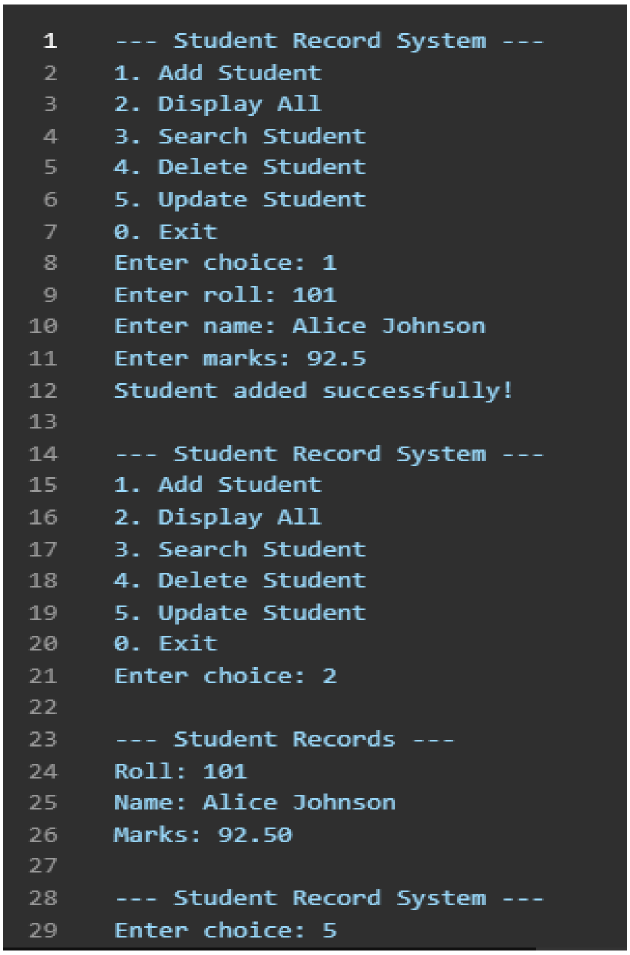
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Student {
   int roll;
   char name[100];
   float marks;
   struct Student *next;
};

struct Student *head = NULL;

// Create a new student node
struct Student* createStudent(int roll, char name[], float marks) {
   struct Student *newStudent = (struct Student*)malloc(sizeof(struct
Student));
```

```
    if (newStudent == NULL) {
        printf("Memory allocation failed!\n");
        exit(1);
    }
    newStudent->roll = roll;
    strcpy(newStudent->name, name);
    newStudent->marks = marks;
    newStudent->next = NULL;
    return newStudent;
}

// Add student to list
void addStudent(int roll, char name[], float marks) {
    struct Student *newStudent = createStudent(roll, name, marks);
    if (head == NULL) {
        head = newStudent;
    } else {
        struct Student *temp = head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newStudent;
    }
    printf("Student added successfully!\n");
}

// Display all students
```

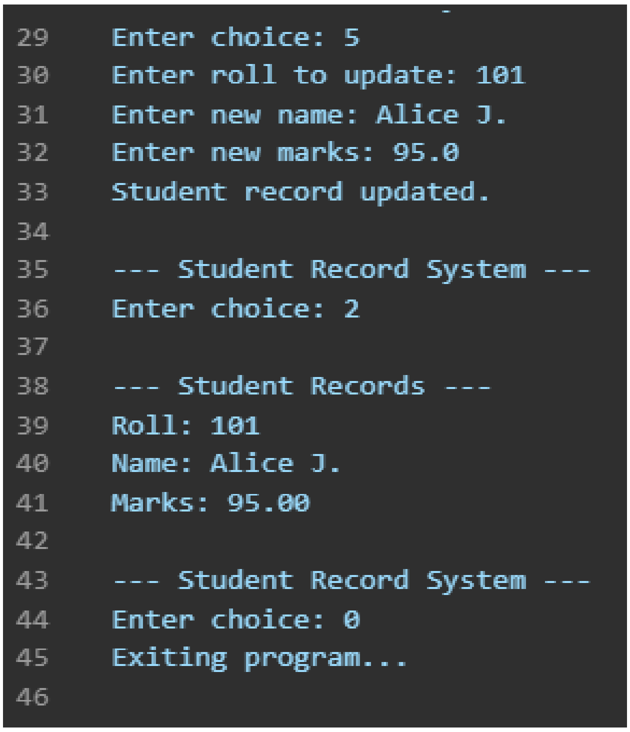
```
void displayStudents() {
    if (head == NULL) {
        printf("No student records available.\n");
        return;
    }

    struct Student *temp = head;
    printf("\n--- Student Records ---\n");
    while (temp != NULL) {
        printf("Roll: %d\nName: %s\nMarks: %.2f\n\n", temp->roll, temp->name,
temp->marks);
        temp = temp->next;
    }
}

// Search student by roll number
void searchStudent(int roll) {
    struct Student *temp = head;
    while (temp != NULL) {
        if (temp->roll == roll) {
            printf("Student Found:\n");
            printf("Roll: %d\nName: %s\nMarks: %.2f\n", temp->roll, temp->name,
temp->marks);
            return;
        }
        temp = temp->next;
    }
    printf("Student with roll %d not found.\n", roll);
}

// Delete student
void deleteStudent(int roll) {
    struct Student *temp = head, *prev = NULL;

    while (temp != NULL && temp->roll != roll) {
        prev = temp;
        temp = temp->next;
    }

    if (temp == NULL) {
        printf("Student with roll %d not found.\n", roll);
        return;
    }

    if (prev == NULL) {
        head = temp->next;
    } else {
        prev->next = temp->next;
    }

    free(temp);
    printf("Student deleted successfully.\n");
}
```

```
// Update student details
void updateStudent(int roll) {
    struct Student *temp = head;
    while (temp != NULL) {
        if (temp->roll == roll) {
            printf("Enter new name: ");
            getchar();  // clear input buffer
            fgets(temp->name, 100, stdin);
            temp->name[strcspn(temp->name, "\n")] = '\0';  // remove newline
            printf("Enter new marks: ");
            scanf("%f", &temp->marks);
            printf("Student record updated.\n");
            return;
        }
        temp = temp->next;
    }
    printf("Student with roll %d not found.\n", roll);
}

// Main function
int main() {
    int choice, roll;
    char name[100];
    float marks;
    do {
        printf("\n--- Student Record System ---\n");
        printf("1. Add Student\n2. Display All\n3. Search Student\n4. Delete Student\n5. Update Student\n0. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter roll: ");
                scanf("%d", &roll);
                getchar();  // clear input buffer
                printf("Enter name: ");
                fgets(name, 100, stdin);
                name[strcspn(name, "\n")] = '\0';  // remove newline
                printf("Enter marks: ");
                scanf("%f", &marks);
                addStudent(roll, name, marks);
                break;
            case 2:
                displayStudents();
                break;
            case 3:
                printf("Enter roll to search: ");
                scanf("%d", &roll);
                searchStudent(roll);
                break;
            case 4:
                printf("Enter roll to delete: ");
                scanf("%d", &roll);
                deleteStudent(roll);
                break;
            case 5:
                printf("Enter roll to update: ");
                scanf("%d", &roll);
                updateStudent(roll);
                break;
            case 0:
                printf("Exiting program...\n");
                break;
            default:
                printf("Invalid choice. Try again.\n");
        }

    } while (choice != 0);

    return 0;
}
```

Output :-



```
--- Student Record System ---
1. Add Student
2. Display All
3. Search Student
4. Delete Student
5. Update Student
0. Exit
Enter choice: 1
Enter roll: 101
Enter name: Alice Johnson
Enter marks: 92.5
Student added successfully!

--- Student Record System ---
1. Add Student
2. Display All
3. Search Student
4. Delete Student
5. Update Student
0. Exit
Enter choice: 2

--- Student Records ---
Roll: 101
Name: Alice Johnson
Marks: 92.50

--- Student Record System ---
Enter choice: 5
```



```
Enter choice: 5
Enter roll to update: 101
Enter new name: Alice J.
Enter new marks: 95.0
Student record updated.

--- Student Record System ---
Enter choice: 2

--- Student Records ---
Roll: 101
Name: Alice J.
Marks: 95.00

--- Student Record System ---
Enter choice: 0
Exiting program...

```