

STUDENT ACADEMIC RECORD SYSTEM

Source code:-

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

struct student {
    int id;
    char name[50];
    char department[50];
    float gpa;
    struct student * prev;
    struct student * next;
};

struct student * head = NULL;

// Function to create a new student node

struct student * createsudent (int id, char name[], char dept, float gpa) {
    struct student * newstudent = (struct student *) malloc (sizeof (struct
    student));
    newstudent->id = id;
    strcpy (newstudent->name, name);
    strcpy (newstudent->department, dept);
    newstudent->gpa = gpa;
    newstudent->prev = NULL;
    newstudent->next = NULL;
    return newstudent;
}
```

// Insert new student

```
void insertstudent (int id, char name[], char dept[], float gpa){
    struct student * newstudent = create student (id, name, dept, gpa);
    if (head == NULL) {
        head = newstudent;
    }
    else {
        struct student * temp = head;
        while (temp->next != NULL)
            temp = temp->next;
        temp->next = newstudent;
        newstudent->prev = temp;
    }
    printf ("Student record inserted.\n");
}
```

// Delete student by ID

```
void deletestudent (int id) {
    struct student * temp = head;
    while (temp != NULL && temp->id != id)
        temp = temp->next;
    if (temp == NULL)
        printf ("Student ID not found.\n");
    return;
    if (temp->prev != NULL) temp->prev->next = temp->next;
    if (temp->next != NULL) temp->next->prev = temp->prev;
    if (temp == head) head = temp->next;
```

```

free(kmp);
printf("student record deleted\n");
}

```

// search by ID or Name

```

void searchStudent(char key[]) {
    struct student *kmp = head;
    int found = 0;
    while (kmp != NULL) {
        char idstr[20];
        sprintf(idstr, "%d", kmp->id);
        if (strcmp(idstr, key) == 0 || strcmp(kmp->name, key) == 0) {
            printf("Found: ID: %d, Name: %s, Dept: %s, GPA: %2f\n",
                kmp->id, kmp->name, kmp->department, kmp->gpa);
            found = 1;
        }
        kmp = kmp->next;
    }
    if (!found) printf("No student found\n");
}

```

// Display all records

```

void displayStudents() {
    struct student *kmp = head;
    if (kmp == NULL) {
        printf("No records found\n");
        return;
    }
}

```

```
printf("In Student Record \n");
```

```
while (temp != NULL) {
```

```
    printf("ID = %d, Name = %s, Dept = %s, GPA = %.2f\n",
```

```
        temp->id, temp->name, temp->department, temp->gpa);
```

```
    temp = temp->next;
```

```
}
```

```
}
```

Reverse display

```
void displayReverse()
```

```
{ if (head == NULL) {
```

```
    printf("No records found\n");
```

```
    return;
```

```
}
```

```
struct student * temp = head;
```

```
while (temp->next != NULL) temp = temp->next;
```

```
printf("In Reverse order Record \n");
```

```
while (temp != NULL) {
```

```
    printf("ID = %d, Name = %s, Dept = %s, GPA = %.2f\n",
```

```
        temp->id, temp->name, temp->department, temp->gpa);
```

```
    temp = temp->prev;
```

clone list

```
struct student * cloneList()
```

```
{ if (head == NULL) return NULL;
```

```
struct student *newhead = NULL, *tail = NULL, *temp = head;
```

```
while(temp != NULL)
```

```
    struct student *newstudent = createstudent(temp->id, temp->name,  
                                                temp->department, temp->gpa);
```

```
    if (newnode == NULL)
```

```
        newhead = newstudent;
```

```
        tail = newhead;
```

```
    else {
```

```
        tail->next = newstudent;
```

```
        newstudent->prev = tail;
```

```
        tail = newstudent;
```

```
    }
```

```
    temp = temp->next;
```

```
}
```

```
printf("List cloned successfully in")
```

```
return newhead;
```

7 Calculate average GPA

```
void averageGPA()
```

```
    struct student *temp = head;
```

```
    if (temp == NULL)
```

```
        printf("No records to calculate GPA in")
```

```
        return;
```

```
    }
```

```
    float sum = 0;
```

```
    int count = 0;
```

```
    while(temp != NULL)
```



```

sum += temp.gpa;
count++;
temp = temp->next;

```

```

2
printf("Average GPA = %f\n", sum/count);

```

Menu driven Program

```

int main() {

```

```

    int choice, id;

```

```

    char name[50], dept[50], key[50];

```

```

    float gpa;

```

```

    struct student *backup = NULL;

```

```

    while (1) {

```

```

        printf("\n == Student Record Management == \n");

```

```

        printf("1. Insert student 2. Delete student 3. Search student 4. Display All\n");

```

```

        printf("5. Display Reverse 6. Clone 7. Average GPA 8. Exit\n");

```

```

        printf("Enter choice");

```

```

        scanf("%d", &choice);

```

```

        getchar();

```

```

        switch (choice) {

```

```

            case 1:

```

```

                printf("Enter ID");

```

```

                scanf("%d", &id);

```

```

                getchar();

```

```

                printf("Enter name");

```

```

fgetc(name, sizeof(name), stdin);
name[strlen(name, "\n")] = 0;
printf("Enter Department: ");
fgetc(dept, sizeof(dept), stdin);
dept[strlen(dept, "\n")] = 0;
printf("Enter GPA: ");
scanf("%f", &gpa);
InsertStudent(id, name, dept, gpa);
break;

```

```

case 2:
    printf("Enter ID to delete");
    scanf("%d", &id);
    deleteStudent(id);
    break;

```

```

case 3:
    printf("Enter ID or Name to search");
    fgetc(key, sizeof(key), stdin);
    key[strlen(key, "\n")] = 0;
    searchStudent(key);
    break;

```

```

case 4:
    displayStudent;

```

```

case 5:
    displayReverse();
    break;

```

```

case 6:
    back p = clone(ptr);

```

break;

case 7:

average = Pa;

break;

case 8:

printf("Exiting... \n");

exit(0);

default:

printf("Invalid choice \n");

}

}

return 0;

}

StudentAcademicRecordSystem.c ×

C: > Users > jo143 > Desktop > StudentAcademicRecordSystem.c > deleteStudent(int)

```
146  int main() {
152      while (1) {
159
160          switch (choice) {
161              case 1:
162                  printf("Enter ID: ");
163                  scanf("%d", &id);
164                  getchar();
165                  printf("Enter Name: ");
166                  fgets(name, sizeof(name), stdin);
167                  name[strcspn(name, "\n")] = 0;
168                  printf("Enter Department: ");
169                  fgets(dept, sizeof(dept), stdin);
170                  dept[strcspn(dept, "\n")] = 0;
171                  printf("Enter GPA: ");
172                  scanf("%f", &gpa);
173                  insertStudent(id, name, dept, gpa);
174                  break;
175              case 2:
176                  printf("Enter ID to delete: ");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

micRecordSystem }

===== Student Record Management System =====

1. Insert Student
2. Delete Student
3. Search Student
4. Display All
5. Display Reverse
6. Clone List
7. Average GPA
8. Exit

Enter choice: 1

Enter ID: Joshua

Enter Name: Enter Department: cse

Enter GPA: 10

Student record inserted.

```
146  int main() {
152      while (1) {
159
160          switch (choice) {
161              case 1:
162                  printf("Enter ID: ");
163                  scanf("%d", &id);
164                  getchar();
165                  printf("Enter Name: ");
166                  fgets(name, sizeof(name), stdin);
167                  name[strcspn(name, "\n")] = 0;
168                  printf("Enter Department: ");
169                  fgets(dept, sizeof(dept), stdin);
170                  dept[strcspn(dept, "\n")] = 0;
171                  printf("Enter GPA: ");
172                  scanf("%f", &gpa);
173                  insertStudent(id, name, dept, gpa);
174                  break;
175              case 2:
176                  printf("Enter ID to delete: ");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

===== Student Record Management System =====

1. Insert Student
2. Delete Student
3. Search Student
4. Display All
5. Display Reverse
6. Clone List
7. Average GPA
8. Exit

Enter choice: 2

Enter ID to delete: Joshua

Student record deleted.

```
146 int main() {
152     while (1) {
159
160         switch (choice) {
161             case 1:
162                 printf("Enter ID: ");
163                 scanf("%d", &id);
164                 getchar();
165                 printf("Enter Name: ");
166                 fgets(name, sizeof(name), stdin);
167                 name[strcspn(name, "\n")] = 0;
168                 printf("Enter Department: ");
169                 fgets(dept, sizeof(dept), stdin);
170                 dept[strcspn(dept, "\n")] = 0;
171                 printf("Enter GPA: ");
172                 scanf("%f", &gpa);
173                 insertStudent(id, name, dept, gpa);
174                 break;
175             case 2:
176                 printf("Enter ID to delete: ");
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

===== Student Record Management System =====

1. Insert Student
2. Delete Student
3. Search Student
4. Display All
5. Display Reverse
6. Clone List
7. Average GPA
8. Exit

Enter choice: Enter ID to delete: Student ID not found.

```

148 int main() {
152     while (1) {
159
160         switch (choice) {
161             case 1:
162                 printf("Enter ID: ");
163                 scanf("%d", &id);
164                 getchar();
165                 printf("Enter Name: ");
166                 fgets(name, sizeof(name), stdin);
167                 name[strcspn(name, "\n")] = 0;
168                 printf("Enter Department: ");
169                 fgets(dept, sizeof(dept), stdin);
170                 dept[strcspn(dept, "\n")] = 0;
171                 printf("Enter GPA: ");
172                 scanf("%f", &gpa);
173                 insertStudent(id, name, dept, gpa);
174                 break;
175             case 2:
176                 printf("Enter ID to delete: ");

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

===== Student Record Management System =====

1. Insert Student
2. Delete Student
3. Search Student
4. Display All
5. Display Reverse
6. Clone List
7. Average GPA
8. Exit

Enter choice: 4

No records found.

C: > Users > jo143 > Desktop > C StudentAcademicRecordSystem.c > main()

```
146  int main() {
152      while (1) {
159
160          switch (choice) {
161              case 1:
162                  printf("Enter ID: ");
163                  scanf("%d", &id);
164                  getchar();
165                  printf("Enter Name: ");
166                  fgets(name, sizeof(name), stdin);
167                  name[strcspn(name, "\n")] = 0;
168                  printf("Enter Department: ");
169                  fgets(dept, sizeof(dept), stdin);
170                  dept[strcspn(dept, "\n")] = 0;
171                  printf("Enter GPA: ");
172                  scanf("%f", &gpa);
173                  insertStudent(id, name, dept, gpa);
174                  break;
175              case 2:
176                  printf("Enter ID to delete: ");
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

==== Student Record Management System ====

1. Insert Student
2. Delete Student
3. Search Student
4. Display All
5. Display Reverse
6. Clone List
7. Average GPA
8. Exit

Enter choice: 5

No records found.

C: > Users > jo143 > Desktop > C StudentAcademicRecordSystem.c > main()

```
146  int main() {
152      while (1) {
159
160          switch (choice) {
161              case 1:
162                  printf("Enter ID: ");
163                  scanf("%d", &id);
164                  getchar();
165                  printf("Enter Name: ");
166                  fgets(name, sizeof(name), stdin);
167                  name[strcspn(name, "\n")] = 0;
168                  printf("Enter Department: ");
169                  fgets(dept, sizeof(dept), stdin);
170                  dept[strcspn(dept, "\n")] = 0;
171                  printf("Enter GPA: ");
172                  scanf("%f", &gpa);
173                  insertStudent(id, name, dept, gpa);
174                  break;
175              case 2:
176                  printf("Enter ID to delete: ");
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

No records found.

==== Student Record Management System ====

1. Insert Student
2. Delete Student
3. Search Student
4. Display All
5. Display Reverse
6. Clone List
7. Average GPA
8. Exit

Enter choice: 7

No records to calculate GPA.

GitHub link

<https://github.com/JoshDev-X/Code-Vault.git>