4)Singly linked list

```
#include  <stdio.h>

#include <stdlib.h>

#include <string.h>


struct Student {

   char usn[20];

   char name[50];

   char programme[20];

   int sem;

   char phoneNumber[15];

   struct Student* next;

};


struct Student* createStudent(char usn[], char name[], char programme[], int sem, char
phoneNumber[]);

struct Student* insertFront(struct Student* head, struct Student* newStudent);

struct Student* insertEnd(struct Student* head, struct Student* newStudent);

void display(struct Student* head);

struct Student* deleteFront(struct Student* head);

struct Student* deleteEnd(struct Student* head);

void freeList(struct Student* head);


int main() {

   struct Student* head = NULL;

   char usn[20], name[50], programme[20], phoneNumber[15];

   int sem;
```

```
    printf("Enter the first student details:\n");
    printf("Enter USN: ");
    scanf("%s", usn);
    printf("Enter Name: ");
    scanf("%s", name);
    printf("Enter Programme: ");
    scanf("%s", programme);
    printf("Enter Semester: ");
    scanf("%d", &sem);
    printf("Enter Phone Number: ");
    scanf("%s", phoneNumber);
    head = createStudent(usn, name, programme, sem, phoneNumber);
    int choice;
    do {
        printf("\n1. Insert at Front\n2. Insert at End\n3. Display\n4. Delete from Front\n5.
Delete from End\n6. Exit\n");
        printf("Enter your choice: ");
        scanf("%d",  &choice);
        switch (choice) {
            case 1: {
                printf("Enter student details:\n");
                printf("Enter USN: ");
                scanf("%s", usn);
                printf("Enter Name: ");
                scanf("%s", name);
                printf("Enter Programme: ");
                scanf("%s", programme);
                printf("Enter Semester: ");
                scanf("%d", &sem);
```

```
            printf("Enter Phone Number: ");
            scanf("%s", phoneNumber);
            struct Student* newStudent = createStudent(usn, name, programme, sem,
phoneNumber);
            head = insertFront(head, newStudent);
            break;
        }
        case 2: {
            printf("Enter student details:\n");
            printf("Enter USN: ");
            scanf("%s", usn);
            printf("Enter Name: ");
            scanf("%s", name);
            printf("Enter Programme: ");
            scanf("%s", programme);
            printf("Enter Semester: ");
            scanf("%d", &sem);
            printf("Enter Phone Number: ");
            scanf("%s", phoneNumber);
            struct Student* newStudent = createStudent(usn, name, programme, sem,
phoneNumber);
            head = insertEnd(head, newStudent);
            break;
        }
        case 3:
            display(head);
            break;
        case 4:
            head = deleteFront(head);
```

```
            break;
        case 5:
            head = deleteEnd(head);
            break;
        case 6:
            freeList(head);
            printf("Exiting the program.\n");
            break;
        default:
            printf("Invalid choice. Please enter a valid option.\n");
    }
  }
  while (choice != 6);
  return 0;
}
struct Student* createStudent(char usn[], char name[], char programme[], int sem, char
phoneNumber[]) {
  struct Student* newStudent = (struct Student*)malloc(sizeof(struct Student));
  strcpy(newStudent->usn, usn);
  strcpy(newStudent->name, name);
  strcpy(newStudent->programme, programme);
  newStudent->sem = sem;
  strcpy(newStudent->phoneNumber, phoneNumber);
  newStudent->next = NULL;
  return newStudent;
}
struct Student* insertFront(struct Student* head, struct Student* newStudent) {
  newStudent->next = head;
  return newStudent;
```

```
}
struct Student* insertEnd(struct Student* head, struct Student* newStudent) {
    if (head == NULL) {
        return newStudent;
    }
    struct Student* temp = head;
    while (temp->next != NULL) {
        temp = temp->next;
    }
    temp->next = newStudent;
    return head;
}
void display(struct Student* head) {
    struct Student* temp = head;
    if (temp == NULL) {
        printf("Student list is empty.\n");
        return;
    }
    printf("\nStudent List:\n");
    while (temp != NULL) {
        printf("USN: %s\n", temp->usn);
        printf("Name: %s\n", temp->name);
        printf("Programme: %s\n", temp->programme);
        printf("Semester: %d\n", temp->sem);
        printf("Phone Number: %s\n", temp->phoneNumber);
        printf("-.....\n");
        temp = temp->next;
    }
```

```
}
struct Student* deleteFront(struct Student* head) {
    if (head == NULL) {
        printf("List is empty. Cannot delete from front.\n");
        return NULL;
    }
    struct Student* temp = head;
    head = head->next;
    free(temp);
    printf("Deleted from front.\n");
    return head;
}
struct Student* deleteEnd(struct Student* head) {
    if (head == NULL) {
        printf("List is empty. Cannot delete from end.\n");
        return NULL;
    }
    if (head->next == NULL) {
        free(head);
        printf("Deleted from end.\n");
        return NULL;
    }
    struct Student* temp = head;
    while (temp->next->next != NULL) {
        temp = temp->next;
    }
    free(temp->next);
    temp->next = NULL;
```

```
    printf("Deleted from end.\n");
    return head;
}
void freeList(struct Student* head) {
    struct Student* temp;
    while (head != NULL) {
        temp = head;
        head = head->next;
        free(temp);
    }
}
```

Output

```
Enter the first student details:
Enter USN: 1
Enter Name: ABC
Enter Programme: Rai
Enter Semester: 3
Enter Phone Number: 1234567890
1. Insert at Front
2. Insert at End
3. Display
4. Delete from Front
5. Delete from End
6. Exit
Enter your choice: 1
Enter student details:
Enter USN: 2
Enter Name: NBS
```