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

typedef struct node {
    char USN[10], name[20], branch[10];
    int sem;
    long int ph;
    struct node *link;
} *NODE;

// Function to allocate a new node
NODE getnode() {
    NODE temp = (NODE)malloc(sizeof(struct node));
    if (temp == NULL) {
        printf("Memory allocation failed\n");
        exit(1);
    }
    temp->link = NULL;
    return temp;
}

NODE ins_front(NODE first) {
    NODE newNode = getnode();
    printf("Enter USN, Name, Branch, Sem and Phone of the student:\n");
    scanf("%s %s %s %d %ld", newNode->USN, newNode->name, newNode->branch, &newNode->sem, &newNode->ph);

    newNode->link = first;
    return newNode;
}

NODE ins_rear(NODE first) {
    NODE temp = getnode();
    printf("Enter USN, Name, Branch, Sem and Phone of the student:\n");
    scanf("%s %s %s %d %ld", temp->USN, temp->name, temp->branch, &temp->sem, &temp->ph);

    if (first == NULL)
        return temp;

    NODE cur = first;
    while (cur->link != NULL)
        cur = cur->link;

    cur->link = temp;
    return first;
}

NODE del_front(NODE first) {
    if (first == NULL) {
        printf("SLL is empty\n");
        return NULL;
    }
```

```
52
53      printf("Information to be deleted is ... \n");
54      printf("%s\t%s\t%s\t%d\t%ld\n", first→USN, first→name, first→branch,
    first→sem, first→ph);
55
56      NODE temp = first→link;
57
58      free(first);
59      return temp;
60  }
61
62  NODE del_rear(NODE first) {
63      if (first == NULL) {
64          printf("SLL is empty\n");
65          return NULL;
66      }
67
68      if (first→link == NULL) {
69          printf("Information to be deleted is ... \n");
70          printf("%s\t%s\t%s\t%d\t%ld\n", first→USN, first→name, first-
    >branch, first→sem, first→ph);
71          free(first);
72          return NULL;
73      }
74
75      NODE temp = first, prev = NULL;
76
77      while (temp→link ≠ NULL) {
78          prev = temp;
79          temp = temp→link;
80      }
81
82      printf("Information to be deleted is ... \n");
83      printf("%s\t%s\t%s\t%d\t%ld\n", temp→USN, temp→name, temp→branch,
    temp→sem, temp→ph);
84      free(temp);
85      prev→link = NULL;
86
87      return first;
88  }
89
90  NODE create(NODE first)
91  {
92      int n, i;
93      printf("Enter the number of students: ");
94      scanf("%d", &n);
95
96      for (i = 0; i < n; i++)
97          first = ins_front(first);
98
99      return first;
100 }
101
102 void status(NODE first)
103 {
```

```
104     int count = 0;
105
106     if (first == NULL) {
107         printf("SLL is empty\n");
108         return;
109     }
110
111     for (; first != NULL; first = first->link, count++)
112         printf("Number of nodes in SLL: %d\n", count);
113 }
114
115 void display(NODE first) {
116     if (first == NULL) {
117         printf("SLL is empty\n");
118         return;
119     }
120
121     printf("Contents of the list:\n");
122     while (first != NULL) {
123         printf("%s\t%s\t%s\t%d\t%ld\n", first->USN, first->name, first-
    >branch, first->sem, first->ph);
124         first = first->link;
125     }
126 }
127 int main() {
128     NODE first = NULL;
129     int ch;
130
131     while (1) {
132         printf("\n1. Create N students\n2. Status of SLL\n");
133         printf("3. Insert front\n4. Insert rear\n5. Delete front\n");
134         printf("6. Delete rear\n7. Display\n8. Exit\n");
135         printf("Enter your choice: ");
136         scanf("%d", &ch);
137
138         switch (ch) {
139             case 1: first = create(first); break;
140             case 2: status(first); break;
141             case 3: first = ins_front(first); break;
142             case 4: first = ins_rear(first); break;
143             case 5: first = del_front(first); break;
144             case 6: first = del_rear(first); break;
145             case 7: display(first); break;
146             case 8: exit(0);
147             default: printf("Invalid choice! Try again.\n");
148         }
149     }
150     return 0;
151 }
152
```