

```

1  /*
2  7. Design, Develop and Implement a menu driven Program in C for the following
3      operations on Singly Linked List (SLL) of Student Data with the fields:
4      USN, Name, Branch, Sem, PhNo
5  a. Create a SLL of N Students Data by using front insertion.
6  b. Display the status of SLL and count the number of nodes in it.
7  c. Perform Insertion/Deletion at End of SLL.
8  d. Perform Insertion/Deletion at Front of SLL(Demonstration of stack)
9  e. Exit
10 */
11
12 /*
13 We create a structure to make sure different data types are used.
14 usn, name, branch are of 'char' datatype.
15 sem,phno are of 'int' datatype.
16 It also has a link pointer variable.
17
18 FIRST is a pointer variable that points to the first node in the linked list.
19
20 The program has 10 functions-
21 1. main function
22 2. Creating a node.
23 3. getnode()
24 4. read()
25 5. CreateSLL()
26 6. displaycount()
27 7. insertfront()
28 8. insertend()
29 9. deletefront()
30 10. deleteend()
31
32 getnode() is used to create a new node everytime the function is called.
33 This new node is initialised to NULL.
34 */
35
36 #include<stdio.h>
37 #include<stdlib.h>
38
39 //creating a structure node
40 struct node          // structure to create a NODE of Student info
41 {
42     char usn[20],name[10],branch[5];
43     int sem,phno;
44     struct node *link;
45 };
46
47 typedef struct node * NODE;          //renaming node as NODE
48
49 /*
50 We declare two structure variables temp & FIRST.
51 temp will be used to create the new node.
52 FIRST represents the First node in the Single Linked List.
53 */
54 NODE temp;
55 NODE FIRST=NULL; //FIRST is initialised to NULL.
56
57 void main()
58 {
59     int ch;
60     while(1)
61     {
62         printf("1- Create SLL\n");
63         printf("2- Display SLL\n");
64         printf("3- Insertfront\n");
65         printf("4- Insert end\n");
66         printf("5- Delete front\n");

```

```

67         printf("6- Delete end\n");
68         printf("7- EXIT\n");
69
70         printf("Enter your choice\n");
71         scanf("%d",&ch);
72
73         switch(ch)
74         {
75             case 1:createSLL();
76                 break;
77             case 2:displaycount();
78                 break;
79             case 3:insertfront();
80                 break;
81             case 4:insertend();
82                 break;
83             case 5:deletefront();
84                 break;
85             case 6:deleteend();
86                 break;
87             case 7:return;
88             default:printf("\n Invalid choice\n");
89         }
90     }
91 } //end of main function.
92
93 /*
94 getnode() function is used to create nodes in the SLL.
95 The memory is allocated dynamically using malloc() function.
96 */
97 NODE getnode() //to create a linked list.
98 {
99     NODE x; //Create a node x
100    x=(NODE)malloc(sizeof(struct node)); //dynamically allocate size
101    x->link=NULL; //create a node that doesn't have a next node.
102    return x;
103 } //end of getnode() function.
104
105 void read() // Read student details
106 {
107     temp=getnode(); //create a new node and put following details
108
109     printf("Enter USN:\n");
110     scanf("%s",temp->usn);
111
112     printf("Enter Name\n");
113     scanf("%s",temp->name);
114
115     printf("Enter Branch\n");
116     scanf("%s",temp->branch);
117
118     printf("Enter Semester\n");
119     scanf("%d",&temp->sem);
120
121     printf("Enter phno:\n");
122     scanf("%d",&temp->phno);
123 } //end of read function
124
125 /*
126 Creating a SLL of N Students data by using front insertion.
127 First we check whether the list is empty.
128 If the list is empty, the newnode inserted will itself become first node.
129 Otherwise we insert the node at first position
130 */
131 void createSLL()
132 {

```

```

133     int i,n;
134
135     printf("Enter the number of students\n");
136     scanf("%d",&n);
137
138     for(i=1;i<=n;i++) //i=1 represents 1st student.
139     {
140         printf("Enter the details of student %d\n",i);
141         read(); //function call to read the student details.
142
143         if(FIRST==NULL) //if the first points to NULL
144             FIRST=temp; //make temp as first node.
145         else
146         {
147             temp->link=FIRST;
148             FIRST=temp; //making new node as first node.
149         }
150     }
151 }
152
153 /*
154 Displaying the status of SLL and count the number of nodes in it.
155 count is used to store the number of nodes.
156 First we check if the list is empty.
157 If list is empty, we consider new node as first node.
158 If not, we display the student details until temp points to NULL.
159 NULL means we have reached the end of the Linked List.
160 */
161 void displaycount()
162 {
163     int count=0; //Initially the number of nodes in linked list=0
164     temp=FIRST; //make temp as the first node
165
166     if(FIRST==NULL) // check for empty list
167         printf("Student List is empty\n");
168     else
169     {
170         printf("Student details is:\n");
171         printf("USN\t Name\t Branch\t Sem\t Phno\n");
172
173         while(temp!=NULL) //Till we reach the NULL(last) node.
174         {
175             count++; //Keep incrementing no.of nodes.
176
177             printf("%s\t %s\t %s\t %d\t %d\n",
178                 temp->usn,temp->name,temp->branch,
179                 temp->sem,temp->phno);
180
181             temp=temp->link; //Move ahead from node to node.
182         }
183         printf("The number of nodes is %d\n",count);
184     }
185     return;
186 } //end of displaycount function
187
188
189 /*
190 Performing Insertion at the front of SLL.
191 First we check list is empty.
192 If list is empty, we consider new node as first node.
193 Otherwise we insert at front of linked list.
194 */
195 void insertfront()
196 {
197     printf("Enter the details of student\n");
198     read();

```

```

199
200     if(FIRST == NULL)
201         FIRST=temp;
202     else
203     {
204         temp->link=FIRST;
205         FIRST=temp;
206     }
207 }
208
209 /*
210 Performing Insertion at the end of SLL.
211 First we check for empty list.
212 If list is empty, we consider new node as first node.
213 'last->link!= NULL' is used to reach the last node.
214 Once it reaches the last node, we make that as 'last' node.
215 Then after the last node, we insert newnode using temp.
216 */
217 void insertend()
218 {
219     NODE last=FIRST; //making last node as the first node.
220
221     printf("Enter the details of student\n");
222     read();
223     if(FIRST==NULL)
224         FIRST=temp;
225     else
226     {
227         while(last->link!= NULL) // loop to reach last node
228             last=last->link;
229         last->link = temp;
230     }
231 }
232
233 /*
234 Performing deletion at the front of SLL.
235 First we check for empty list.
236 If not empty, we start to delete from front node of list.
237 Deletion is done based on unique number 'usn'.
238 We make the second node as first node & then delete first node.
239 */
240 void deletefront()
241 {
242     temp=FIRST; //First node is assigned to temp.
243
244     if(FIRST==NULL)
245         printf("List is empty\n");
246     else
247     {
248         printf("deleted element is %s\n",temp->usn);
249         FIRST=FIRST->link; //making second node as first node
250         free(temp); // deleting first node.
251     }
252     return;
253 }
254
255 /*
256 Performing Deletion at the end of SLL.
257 We assign last node to NULL.
258 First we check for empty list.
259 If not empty,we check if linked list has only one node & delete that.
260 If both these conditions are false, we delete node from the end of list.
261 */
262
263 void deleteend()
264 {

```

```

265     NODE last=NULL;           //last node made as NULL.
266
267     temp=FIRST;               //First node is assigned to temp.
268
269     if(FIRST == NULL)
270         printf("List is empty\n");
271
272     else if(FIRST->link == NULL) //Means has only one node.
273     {
274         printf("Deleted element is %s\n",temp->usn);
275         free(FIRST);
276         FIRST=NULL;
277     }
278     else
279     {
280         while(temp->link!=NULL) // loop to reach last node
281         {
282             last=temp;
283             temp=temp->link;    //Reaches last node.
284         }
285         last->link=NULL;
286         printf("Deleted element is %s\n",temp->usn);
287         free(temp);
288     }
289     return;
290 }

```

OUTPUT:

```
1- Create SLL
2- Display SSL
3- Insertfront
4- Insert end
5- Delete front
6- Delete end
7- EXIT
Enter your choice
2
Student List is empty
```

```
1- Create SLL
2- Display SSL
3- Insertfront
4- Insert end
5- Delete front
6- Delete end
7- EXIT
Enter your choice
5
List is empty
```

```
1- Create SLL
2- Display SSL
3- Insertfront
4- Insert end
5- Delete front
6- Delete end
7- EXIT
Enter your choice
6
List is empty
```

```
1- Create SLL
2- Display SSL
3- Insertfront
4- Insert end
5- Delete front
6- Delete end
7- EXIT
Enter your choice
1
Enter the number of students
3
Enter the details of student 1
Enter USN:
111
Enter Name
```

ABC
Enter Branch
CSE
Enter Semester
3
Enter phno:
9890
Enter the details of student 2
Enter USN:
222
Enter Name
XYZ
Enter Branch
ISE
Enter Semester
5
Enter phno:
7650
Enter the details of student 3
Enter USN:
333
Enter Name
PQR
Enter Branch
ECE
Enter Semester
7
Enter phno:
6789

1- Create SLL
2- Display SSL
3- Insertfront
4- Insert end
5- Delete front
6- Delete end
7- EXIT
Enter your choice
2

Student details is:

USN	Name	Branch	Sem	Phno
333	PQR	ECE	7	6789
222	XYZ	ISE	5	7650
111	ABC	CSE	3	9890

The number of nodes is 3

1- Create SLL
2- Display SSL
3- Insertfront
4- Insert end

5- Delete front

6- Delete end

7- EXIT

Enter your choice

3

Enter the details of student

Enter USN:

444

Enter Name

JKL

Enter Branch

ME

Enter Semester

4

Enter phno:

9234

1- Create SLL

2- Display SSL

3- Insertfront

4- Insert end

5- Delete front

6- Delete end

7- EXIT

Enter your choice

3

Student details is:

USN	Name	Branch	Sem	Phno
444	JKL	ME	4	9234
333	PQR	ECE	7	6789
222	XYZ	ISE	5	7650
111	ABC	CSE	3	9890

The number of nodes is 4

1- Create SLL

2- Display SSL

3- Insertfront

4- Insert end

5- Delete front

6- Delete end

7- EXIT

Enter your choice

4

Enter the details of student

Enter USN:

666

Enter Name

DEF

Enter Branch

CV

Enter Semester

2

Enter phno:

6543

- 1- Create SLL
- 2- Display SSL
- 3- Insertfront
- 4- Insert end
- 5- Delete front
- 6- Delete end
- 7- EXIT

Enter your choice

2

Student details is:

USN	Name	Branch	Sem	Phno
444	JKL	ME	4	9234
333	PQR	ECE	7	6789
222	XYZ	ISE	5	7650
111	ABC	CSE	3	9890
666	DEF	CV	2	6543

The number of nodes is 5

- 1- Create SLL
- 2- Display SSL
- 3- Insertfront
- 4- Insert end
- 5- Delete front
- 6- Delete end
- 7- EXIT

Enter your choice

5

deleted element is 444

- 1- Create SLL
- 2- Display SSL
- 3- Insertfront
- 4- Insert end
- 5- Delete front
- 6- Delete end
- 7- EXIT

Enter your choice

6

Deleted element is 666

- 1- Create SLL
- 2- Display SSL
- 3- Insertfront
- 4- Insert end
- 5- Delete front

6- Delete end

7- EXIT

Enter your choice

2

Student details is:

USN	Name	Branch	Sem	Phno
333	PQR	ECE	7	6789
222	XYZ	ISE	5	7650
111	ABC	CSE	3	9890

The number of nodes is 3

1- Create SLL

2- Display SSL

3- Insertfront

4- Insert end

5- Delete front

6- Delete end

7- EXIT

Enter your choice

7