

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  // #define SIZE 20
4
5  typedef struct node {
6      char SSN[10], name[20], dept[30], desig[30];
7      float sal;
8      long int ph;
9      struct node *llink, *rlink;
10 } *NODE;
11
12 // Function to allocate a new node
13 NODE getnode() {
14     NODE temp = (NODE)malloc(sizeof(struct node));
15     if (temp == NULL) {
16         printf("Memory allocation failed\n");
17         exit(1);
18     }
19     temp->llink = temp->rlink = NULL;
20     return temp;
21 }
22
23 NODE ins_front(NODE first) {
24     NODE temp = getnode();
25     printf("Enter SSN, Name, Dept, Desig, Salary, Phone of the Employee:\n");
26     scanf("%s%s%s%s%f%ld", temp->SSN, temp->name, temp->dept, temp->desig,
27 &temp->sal, &temp->ph);
28
29     temp->rlink = first;
30     if (first != NULL)
31         first->llink = temp;
32
33     return temp;
34 }
35
36 NODE ins_rear(NODE first) {
37     NODE temp = getnode();
38     printf("Enter SSN, Name, Dept, Desig, Salary, Phone of the Employee:\n");
39     scanf("%s%s%s%s%f%ld", temp->SSN, temp->name, temp->dept, temp->desig,
40 &temp->sal, &temp->ph);
41
42     if (first == NULL)
43         return temp;
44
45     NODE cur = first;
46     while (cur->rlink != NULL)
47         cur = cur->rlink;
48
49     cur->rlink = temp;
50     temp->llink = cur;
51     return first;
52 }
```

```
52  NODE del_front(NODE first) {
53      if (first == NULL) {
54          printf("DLL is empty\n");
55          return NULL;
56      }
57
58      printf("Information to be deleted is ... \n");
59      printf("%s\t%s\t%s\t%s\t%f\t%d\n", first->SSN, first->name, first->dept,
60      first->desig, first->sal, first->ph);
61
62      NODE temp = first->rlink;
63      if (temp != NULL)
64          temp->llink = NULL;
65
66      free(first);
67      return temp;
68  }
69
70  NODE del_rear(NODE first) {
71      if (first == NULL) {
72          printf("DLL is empty\n");
73          return NULL;
74      }
75
76      NODE temp = first, prev = NULL;
77
78      while (temp->rlink != NULL) {
79          prev = temp;
80          temp = temp->rlink;
81      }
82
83      printf("Information to be deleted is ... \n");
84      printf("%s\t%s\t%s\t%s\t%f\t%d\n", temp->SSN, temp->name, temp->dept,
85      temp->desig, temp->sal, temp->ph);
86      free(temp);
87
88      if (prev != NULL) {
89          prev->rlink = NULL;
90          return first;
91      }
92
93      return NULL;
94  }
95
96  NODE create(NODE first) {
97      int n;
98      printf("Enter the number of employees: ");
99      scanf("%d", &n);
100
101      for (int i = 0; i < n; i++)
102          first = ins_rear(first);
103
104      return first;
105  }
```

```
105 void status(NODE first) {
106     int count = 0;
107
108     if (first == NULL) {
109         printf("DLL is empty\n");
110         return;
111     }
112
113     for (; first != NULL; first = first->rlink, count++);
114     printf("Number of nodes in DLL is %d\n", count);
115 }
116
117 void display(NODE first) {
118     if (first == NULL) {
119         printf("Contents of DLL are empty\n");
120         return;
121     }
122
123     printf("Contents of the list from FIRST → LAST:\n");
124     NODE cur = first;
125     while (cur != NULL) {
126         printf("%s\t%s\t%s\t%s\t%f\t%ld\n", cur->SSN, cur->name, cur->dept,
127             cur->desig, cur->sal, cur->ph);
128         cur = cur->rlink;
129     }
130
131     printf("Contents of the list from LAST → FIRST:\n");
132     // Move to the last node
133     while (first->rlink != NULL)
134         first = first->rlink;
135
136     // Print from last to first
137     while (first != NULL) {
138         printf("%s\t%s\t%s\t%s\t%f\t%ld\n", first->SSN, first->name, first->
139             >dept, first->desig, first->sal, first->ph);
140         first = first->llink;
141     }
142 }
143
144 int main() {
145     NODE first = NULL;
146     int ch;
147
148     for (;;) {
149         printf("1. Create N employees\n2. Status of DLL\n");
150         printf("3. Insert front\n4. Insert rear\n5. Delete front\n");
151         printf("6. Delete rear\n7. Display\n8. Exit\n");
152         scanf("%d", &ch);
153         switch (ch) {
154             case 1: first = create(first); break;
155             case 2: status(first); break;
156             case 3: first = ins_front(first); break;
157             case 4: first = ins_rear(first); break;
158             case 5: first = del_front(first); break;
159             case 6: first = del_rear(first); break;
```

```
158         case 7: display(first); break;
159         case 8: exit(0);
160         default: printf("Invalid choice! Try again.\n");
161     }
162 }
163 }
164
```