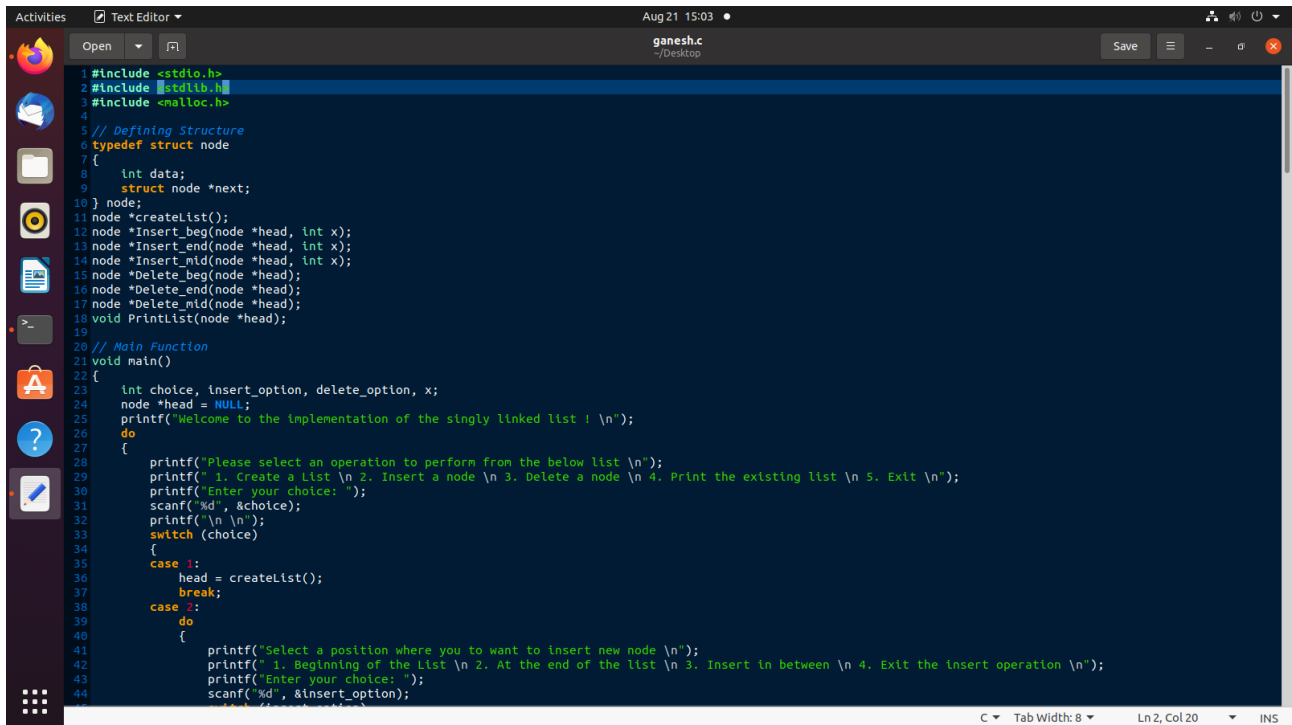
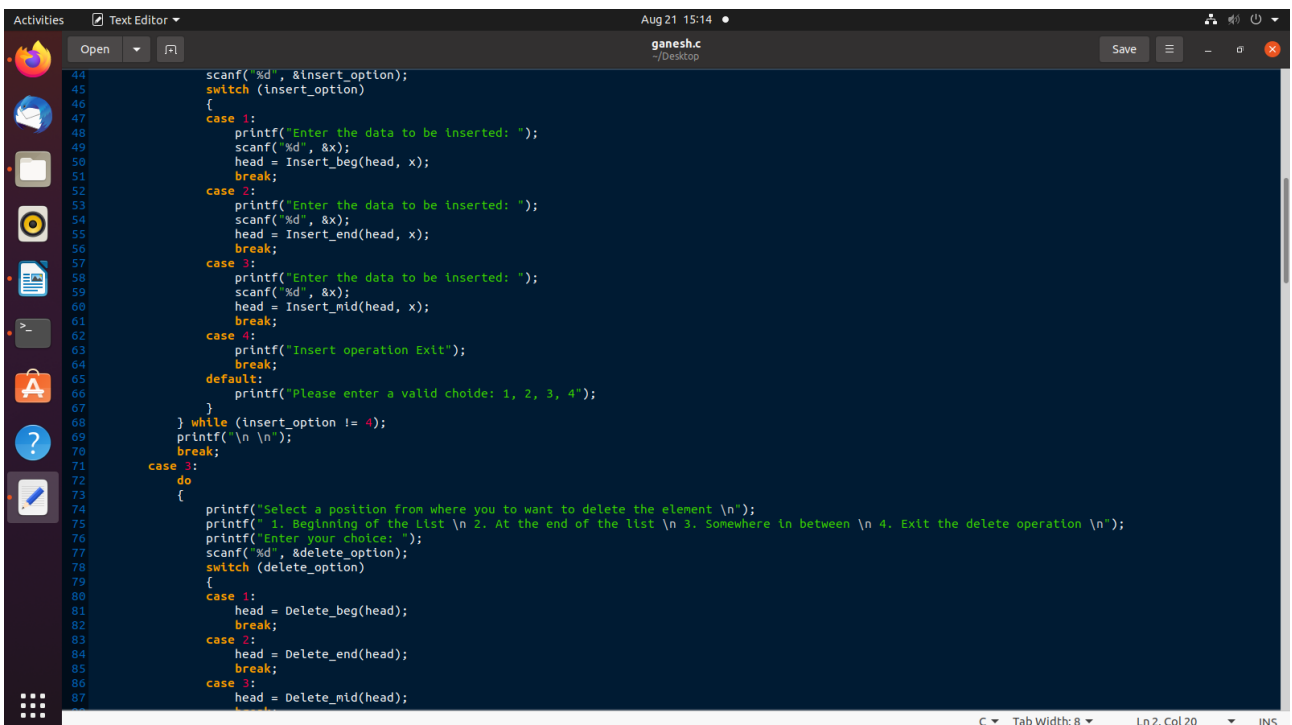


Name: ShreeGanesh Vishwakarma  
Class: SYIT      Experiment No. 5

Program:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <malloc.h>
4
5 // Defining Structure
6 typedef struct node
7 {
8     int data;
9     struct node *next;
10 } node;
11 node *createList();
12 node *Insert_beg(node *head, int x);
13 node *Insert_end(node *head, int x);
14 node *Insert_mid(node *head, int x);
15 node *Delete_beg(node *head);
16 node *Delete_end(node *head);
17 node *Delete_mid(node *head);
18 void PrintList(node *head);
19
20 // Main Function
21 void main()
22 {
23     int choice, insert_option, delete_option, x;
24     node *head = NULL;
25     printf("Welcome to the implementation of the singly linked list ! \n");
26     do
27     {
28         printf("Please select an operation to perform from the below list \n");
29         printf("1. Create a List \n 2. Insert a node \n 3. Delete a node \n 4. Print the existing list \n 5. Exit \n");
30         printf("Enter your choice: ");
31         scanf("%d", &choice);
32         printf("\n \n");
33         switch (choice)
34         {
35             case 1:
36                 head = createList();
37                 break;
38             case 2:
39                 do
40                 {
41                     printf("Select a position where you want to insert new node \n");
42                     printf("1. Beginning of the List \n 2. At the end of the list \n 3. Insert in between \n 4. Exit the insert operation \n");
43                     printf("Enter your choice: ");
44                     scanf("%d", &insert_option);
```



```
44         scanf("%d", &insert_option);
45         switch (insert_option)
46         {
47             case 1:
48                 printf("Enter the data to be inserted: ");
49                 scanf("%d", &x);
50                 head = Insert_beg(head, x);
51                 break;
52             case 2:
53                 printf("Enter the data to be inserted: ");
54                 scanf("%d", &x);
55                 head = Insert_end(head, x);
56                 break;
57             case 3:
58                 printf("Enter the data to be inserted: ");
59                 scanf("%d", &x);
60                 head = Insert_mid(head, x);
61                 break;
62             case 4:
63                 printf("Insert operation Exit");
64                 break;
65             default:
66                 printf("Please enter a valid choide: 1, 2, 3, 4");
67         }
68     } while (insert_option != 4);
69     printf("\n \n");
70     break;
71     case 3:
72     do
73     {
74         printf("Select a position from where you want to delete the element \n");
75         printf("1. Beginning of the List \n 2. At the end of the list \n 3. Somewhere in between \n 4. Exit the delete operation \n");
76         printf("Enter your choice: ");
77         scanf("%d", &delete_option);
78         switch (delete_option)
79         {
80             case 1:
81                 head = Delete_beg(head);
82                 break;
83             case 2:
84                 head = Delete_end(head);
85                 break;
86             case 3:
87                 head = Delete_mid(head);
88                 break;
```

```
Activities Text Editor Aug 21 15:03
ganesh.c ~/Desktop Save
87 head = Delete_mid(head);
88 break;
89 case 4:
90 printf("Delete Operation Exit");
91 break;
92 default:
93 printf("Please enter a valid choide: 1, 2, 3, 4");
94 } while (delete_option != 4);
95 printf("\n\n");
96 break;
97 case 4:
98 PrintList(head);
99 break;
100 case 5:
101 printf("Exit: Program Finished !!");
102 break;
103 default:
104 printf("Please enter a valid choide: 1, 2, 3, 4, 5");
105 } while (choice != 5);
106 }
107 } while (choice != 5);
108 }
109
110 // Function to create List
111 node *createList()
112 {
113 node *head, *p;
114 int i, n;
115 head = NULL;
116 printf("Enter the number of nodes: ");
117 scanf("%d", &n);
118 printf("Enter the data: ");
119 for (i = 0; i <= n - 1; i++)
120 {
121 if (head == NULL)
122 {
123 p = head = (node *)malloc(sizeof(node));
124 }
125 else
126 {
127 p->next = (node *)malloc(sizeof(node));
128 p = p->next;
129 }
130 p->next = NULL;
131 scanf("%d", &(p->data));
132 }
```

```
Activities Text Editor Aug 21 15:03
ganesh.c ~/Desktop Save
130 p->next = NULL;
131 scanf("%d", &(p->data));
132 }
133 printf("\n\n");
134 return (head);
135 }
136
137 // Function to insert element
138 node *Insert_beg(node *head, int x)
139 {
140 node *p;
141 p = (node *)malloc(sizeof(node));
142 p->data = x;
143 p->next = head;
144 head = p;
145 return (head);
146 }
147 node *Insert_end(node *head, int x)
148 {
149 node *p, *q;
150 p = (node *)malloc(sizeof(node));
151 p->data = x;
152 p->next = NULL;
153 if (head == NULL)
154 return (p);
155 for (q = head; q->next != NULL; q = q->next)
156 ;
157 q->next = p;
158 return (head);
159 }
160 node *Insert_mid(node *head, int x)
161 {
162 node *p, *q;
163 int y;
164 p = (node *)malloc(sizeof(node));
165 p->data = x;
166 p->next = NULL;
167 printf("After which element you want to insert the new element ?");
168 scanf("%d", &y);
169 for (q = head; q != NULL && q->data != y; q = q->next)
170 ;
171 if (q != NULL)
172 {
173 p->next = q->next;
174 q->next = p;
175 }
```

Activities Text Editor Aug 21 15:09 ganesh.c ~/Desktop Save

```
173     p->next = q->next;
174     q->next = p;
175 }
176 else
177     printf("ERROR !! Data Not Found");
178     return (head);
179 }
180
181 // Function to delete element
182 node *Delete_beg(node *head)
183 {
184     node *p, *q;
185     if (head == NULL)
186     {
187         printf("Empty Linked List");
188         return (head);
189     }
190     p = head;
191     head = head->next;
192     free(p);
193     return (head);
194 }
195 node *Delete_end(node *head)
196 {
197     node *p, *q;
198     if (head == NULL)
199     {
200         printf("Empty Linked List");
201         return (head);
202     }
203     p = head;
204     if (head->next == NULL)
205     {
206         head = NULL;
207         free(p);
208         return (head);
209     }
210     for (q = head; q->next->next != NULL; q = q->next)
211         p = q->next;
212     q->next = NULL;
213     free(p);
214     return (head);
215 }
216 node *Delete_mid(node *head)
```

C Tab Width: 8 Ln 2, Col 20 INS

Activities Text Editor Aug 21 15:09 ganesh.c ~/Desktop Save

```
214     return (head);
215 }
216 node *Delete_mid(node *head)
217 {
218     node *p, *q;
219     int x, i;
220     if (head == NULL)
221     {
222         printf("Empty Linked List");
223         return (head);
224     }
225     printf("Enter the data to be deleted: ");
226     scanf("%d", &x);
227     if (head->data == x)
228     {
229         p = head;
230         head = head->next;
231         free(p);
232         return (head);
233     }
234     for (q = head; q->next->data != x && q->next != NULL; q = q->next)
235         if (q->next == NULL)
236         {
237             printf("ERROR !! Data Not Found");
238             return (head);
239         }
240     p = q->next;
241     q->next = q->next->next;
242     free(p);
243     return (head);
244 }
245
246 // Function to print the existing list
247 void PrintList(node *head)
248 {
249     node *p;
250     printf("[ ");
251     for (p = head; p != NULL; p = p->next)
252     {
253         printf("%d \t", p->data);
254     }
255     printf(" ]");
256     printf("\n \n");
257 }
```

C Tab Width: 8 Ln 2, Col 20 INS

## Output:

```
Activities Terminal Aug 21 15:02 itadmin@itadmin-HP-ProDesk-400-G7-Microtower-PC: ~/Desktop
itadmin@itadmin-HP-ProDesk-400-G7-Microtower-PC:~/Desktop$ gcc ganesh.c
itadmin@itadmin-HP-ProDesk-400-G7-Microtower-PC:~/Desktop$ ./a.out ganesh.c
Welcome to the implementation of the singly linked list !
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 1
Enter the number of nodes: 4
Enter the data: 4
5
5
1
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 2
Select a position where you to want to insert new node
1. Beginning of the List
2. At the end of the list
3. Insert in between
4. Exit the insert operation
Enter your choice: 1
Enter the data to be inserted: 6
Select a position where you to want to insert new node
1. Beginning of the List
2. At the end of the list
3. Insert in between
4. Exit the insert operation
Enter your choice: 2
Enter the data to be inserted: 5
Select a position where you to want to insert new node
1. Beginning of the List
2. At the end of the list
3. Insert in between
```

```
Activities Terminal Aug 21 15:03 itadmin@itadmin-HP-ProDesk-400-G7-Microtower-PC: ~/Desktop
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 2
Select a position where you to want to insert new node
1. Beginning of the List
2. At the end of the list
3. Insert in between
4. Exit the insert operation
Enter your choice: 1
Enter the data to be inserted: 6
Select a position where you to want to insert new node
1. Beginning of the List
2. At the end of the list
3. Insert in between
4. Exit the insert operation
Enter your choice: 2
Enter the data to be inserted: 5
Select a position where you to want to insert new node
1. Beginning of the List
2. At the end of the list
3. Insert in between
4. Exit the insert operation
Enter your choice: 4
Insert operation Exit
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
5. Exit
Enter your choice: 4
[ 6 4 5 5 1 5 ]
Please select an operation to perform from the below list
1. Create a List
2. Insert a node
3. Delete a node
4. Print the existing list
```