```
1
    /* creation of singly linked list & Operations On Singly Linked List
 2
        ->insertion
 3
             ->at begning
 4
             ->at end
 5
            ->at specificposition
 6
        -> Deletion
 7
            ->at begning
8
            ->at end
9
            ->at specificposition
10
        ->find the length of list
11
        ->reverce the list
12 // Created By SMIT
1.3
   * /
   #include<stdio.h>
14
15
    #include<conio.h>
16
    #include<stdlib.h>
17
18
19
   //Global Declaractions
20 int count=0;
21 struct node{
22
        int data;
23
        struct node *next;
24 };
    struct node *head=0,*temp; //Global Pointers
25
    void CreateList(); //To Create Link LIst
26
27
    void InsAtBeg();    //To Insert Data At Beginig
                        //This Function will Insert Data At Given Position
28
    void InsAtPos();
29
    void InsAtEnd();
                        //To Insert Data At The End Of List
   void DelAtBeg();
30
                        //To delete first Element/Node Of link list
                        //To delete Spasific Node Of Link LIst
31 void DelAtPos();
32
   void DelAtEnd();
                        //To delete Last Node Of List
33 void FindLength(); //To find the length Of Link list
34 void ReverseList(); //To reverse The Link List
35
    void Display();
                       //To Display Link LIst
36
37
    int main(){
38
         int choice;
                             //This Loop Will Run Untill You Give 11 As an Input
39
         while (choice!=11) {
             printf("\n***********MAIN MENU*********\n");
40
41
             //choice Must Beatween 1 To 11
42
            printf("1.Create a list\n");
43
            printf("2.insert at begning\n");
44
            printf("3.insert at n(th) position\n");
45
            printf("4.insert at end\n");
46
            printf("5.delete from begning\n");
47
            printf("6.delete from n(th) position\n");
48
            printf("7.delete from end\n");
49
            printf("8.Find the length\n");
50
            printf("9.Reverse the Linked list\n");
51
            printf("10.Display Linked List\n");
52
            printf("11.exit\n");
53
             scanf("%d", &choice);
54
55
             //To Ran Spacific Function According To choice
56
         switch(choice) {
57
             case 1:
58
                 CreateList();
59
                 break;
60
             case 2:
61
                 InsAtBeq();
62
                break;
63
            case 3:
64
                 InsAtPos();
65
                break;
66
            case 4:
67
                 InsAtEnd();
68
                break;
69
             case 5:
```

```
70
                   DelAtBeq();
 71
                  break;
 72
               case 6:
 73
                   DelAtPos();
 74
                  break;
 75
              case 7:
                  DelAtEnd();
 76
 77
                  break;
 78
              case 8:
 79
                  FindLength();
 80
                  break;
 81
              case 9:
 82
                   ReverseList();
 83
                   break;
 84
              case 10:
 85
                   Display();
 86
                   break;
 87
              case 11:
 88
                   exit(0);
 89
                   break;
 90
              default:
 91
                   printf("\n\n\nEnter valid choice\n\n\n");
 92
          }
 93
      }
 94
          return 0;
      // Created By SMIT
 95
 96
 97
 98
      //This Function Will Create A Linked List
 99
     void CreateList(){
100
     struct node *newnode;
101
     int choice=1;
102
      while(choice) {
103
      newnode=(struct node*)malloc(sizeof(struct node));
104
      printf("Enter Data ");
      scanf("%d", &newnode->data);
105
106
      newnode->next=0;
107
      if (head==0) {
108
      head=temp=newnode;
109
      - }
110
      else{
111
          temp->next=newnode;
112
          temp=newnode;
113
      }
114
      printf("\nPress 0 to exit Press 1 To Add ");
115
      scanf("%d", &choice);
116
      count++;
117
118
      printf("\n\n\n%d nodes created successfully\n\n\n",count);
119
120
121
      //This Function Will Display The List
122
     void Display() {
123
          if (head==0) {
124
              printf("\n\n\nI Think You should Create A list First\n\n\n");
125
          }
126
          else{
127
               temp=head;
128
              printf("\n\n\nThe List is");
129
              while(temp!=0) {
130
                   printf("\n%d\n",temp->data);
131
                   temp=temp->next;
132
               }
133
          }
134
135
136
      //To Insert Node At Begning
137
      void InsAtBeg() {
138
          if (head==0) {
```

```
139
              printf("\n\n\nPlease Create a List First\n\n\n");
140
          }
141
          else{
          struct node *newnode;
142
143
          newnode=(struct node *)malloc(sizeof(struct node));
144
          printf("Enter data");
145
          scanf("%d", &newnode->data);
146
          newnode->next=head;
147
          head=newnode;
148
          1
149
      }
150
1.5.1
      //To Insert Node At spacific Positon
152
      void InsAtPos(){
153
          if (head==0) {
               printf("\n\n\nPlease Create a List First\n\n\n");
154
155
          }
156
          else{
157
               int pos,i=1;
158
              printf("Enter Position");
159
               scanf("%d", &pos);
160
               if (pos>count) {
161
                   printf("Invalid Position");
162
               }
163
               else{
164
                   struct node *newnode;
165
                   temp=head;
166
                   while(i<pos) {</pre>
167
                       temp=temp->next;
168
                       i++;
169
                   }
170
                   newnode=(struct node*)malloc(sizeof(struct node));
171
                   printf("Enter Data");
                   scanf("%d",&newnode->data);
172
173
                   newnode->next=temp->next;
174
                   temp->next=newnode;
175
               }
176
          }
177
      }
178
179
      //To Insert Node At The End Of List
180
      void InsAtEnd() {
181
          if (head==0) {
182
               printf("\n\n\nPlease Create a List First\n\n\n");
183
          1
184
          else{
185
               struct node *newnode;
186
               temp=head;
187
              while(temp->next!=0) {
188
                   temp=temp->next;
189
               }
190
              newnode=(struct node*)malloc(sizeof(struct node));
191
              printf("Enter Data");
192
              scanf("%d", &newnode->data);
193
              newnode->next=0;
194
               temp->next=newnode;
195
          }
196
      }
197
198
      //To Delete First Node
199
      void DelAtBeq() {
200
          if (head==0) {
201
               printf("\n\n\nPlease Create a List First\n\n\n");
202
          }
203
          else{
204
               temp=head;
205
              head=temp->next;
206
               free (temp);
207
               printf("\n\n\nDeleted successfully..\n\n\n");
```

```
208
          }
209
      }
210
      // Created By Smit
211
      //To Delete Node AT Spacific Position
212
      void DelAtPos() {
213
          int pos, i=1;
214
          if (head==0) {
               printf("\n\n\nPlease Create A list First\n\n\n");
215
216
217
          else{
218
               struct node *deletethis;
219
               temp=head;
220
               printf("Enter Position ");
               scanf("%d",&pos);
221
222
               if (pos>count) {
                   printf("\n\nInvalid Position\n\n");
223
224
               }
               else{
225
226
                   while(i<pos-1){</pre>
227
                       temp=temp->next;
228
                       i++;
229
                   }
230
                   deletethis=temp->next;
231
                   temp->next=deletethis->next;
232
                   free (deletethis);
233
                   printf("\n\n\nDeleted successfully...\n\n\n");
234
               }
235
          }
236
      }
237
238
      //To Delete Last Node
239
      void DelAtEnd() {
240
          if (head==0) {
               printf("\n\n\nPlease Create A List First\n\n\n");
241
242
          1
243
          else{
244
               struct node *prev;
245
               temp=head;
246
               while(temp->next!=0) {
247
                   prev=temp;
248
                   temp=temp->next;
249
               }
250
               free (temp);
251
               prev->next=0;
252
               printf("\n\n\nDeleted Successfully...\n\n\n");
253
          }
254
      }
255
256
      //To Find The Lenth Of List
257
      void FindLength(){
258
          if (head==0) {
259
               printf("\n\n\nYou can't find length Without Creating List\n\n\n");
260
          }
261
          else{
262
          temp=head;
263
          count=0;
264
          while(temp!=0)
265
               temp=temp->next;
266
               count++;
267
268
          printf("\n\n\nThe Length Of Linked List Is %d \n\n\n",count);
269
          }
270
      }
271
272
      //To Reverse The List
273
      void ReverseList(){
274
          if (head==0) {
275
               printf("\n\n\nThere Is No List To Reverse\n\n\n");
276
```

```
277
         else{
278
            struct node *previousnode,*currentnode,*nextnode;
279
            previousnode=0;
280
             currentnode=nextnode=head;
281
             while (nextnode!=0) {
282
             nextnode=nextnode->next;
283
             currentnode->next=previousnode;
284
             previousnode=currentnode;
285
             currentnode=nextnode;
286
             }
287
             head=previousnode;
             printf("\n\n\nList Reversed Successfully..\n\n\n");
288
289
         }
290
291
     // Created By SMIT
292
```