

```
1  #include<stdio.h>
2  #include<string.h>
3  #include<time.h>
4  #include<math.h>
5  #include<stdlib.h>
6  #include<ctype.h>
7  #include<conio.h>
8
9  struct Data{
10     int angka;
11
12     Data *next;
13 } *head = NULL, *tail = NULL;
14
15 void pushHead(int angka){
16
17     Data *node = (Data *) malloc(sizeof(Data));
18
19     //(*node).angka = angka;
20     node->angka = angka;
21
22     if(head==NULL){
23         head = tail = node;
24     } else{
25         node->next = head;
26         head = node;
27     }
28     tail->next = NULL;
29 }
30
31 void pushTail(int angka){
32
33     Data *node = (Data *) malloc(sizeof(Data));
34
35     node->angka = angka;
36
37     if(head==NULL){
38         head = tail = node;
39     } else{
40         tail->next = node;
41         tail = node;
42     }
43     tail->next = NULL;
44 }
45
46 void push(int a){
47     if(head==NULL){
48         pushHead(a);
49     }else{
50         if(a < head->angka){
51             pushHead(a);
52             //if angka < head
53         }else if (a > tail->angka){
54             pushTail(a);
55             //if angka > head
56         }else{
```

```
57         //if head < angka < tail (between)
58         Data *temp = head;
59         while(temp->next->angka < a){
60             temp = temp->next;
61         }
62         Data *node = (Data *) malloc(sizeof(Data));
63         node->angka = a;
64         node->next = temp->next;
65         temp->next = node;
66     }
67 }
68 }
69
70 void pop(int a){
71     //check if there is data on the list
72     if(head){//if there is data
73         if(head->angka == a){
74             //if angka to delete is on head
75             if(head==tail){//if data is left 1 on the list
76                 free(head);
77                 head = tail = NULL;
78             } else{//if data > 1 on the list
79                 Data *temp = head;
80                 head = head->next;
81                 free(temp);
82             }
83         } else if(tail->angka == a) { //if angka to delete is on tail
84             Data *temp = head;
85             while(temp->next != tail){
86                 temp = temp->next;
87             }
88             free(tail);
89             tail = temp;
90             tail->next = NULL;
91         } else {
92             //if angka to delete is in-between, middle
93             Data *temp = head;
94             while(temp != NULL && temp -> angka != a){
95                 temp = temp ->next;
96             }
97             if(temp!=NULL){//if data being search is found
98                 Data *temp2 = head;
99                 while(temp2->next != temp){
100                     temp2= temp2 ->next;
101                 }
102                 temp2->next = temp->next;
103                 free(temp);
104             }
105         }
106     }
107 }else{//if data not on the list
108     printf("Data Kosong");
109 }
110 }
111 }
112 }
```

```
113 void viewList(){
114     Data *temp;
115     temp = head;
116     while(temp!=NULL){
117         printf("%d->", temp->angka);
118         temp = temp->next;
119     }
120     printf("NULL");
121 }
122
123 int main(){
124     //linked list
125     //single and double
126     /*pushHead(6);
127     pushHead(8);*/
128     /*pushHead(9);
129     pushHead(12);
130     pushHead(17);*/
131     /*pushTail(7);*/
132     push(5);
133     push(12);
134     push(30);
135     push(6);
136     push(14);
137
138     pop(5);
139     pop(14);
140     pop(30);
141
142
143     viewList();
144
145     getchar();
146     return 0;
147 }
```