

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  struct elm_jual;
6  struct elm_sales;
7
8  typedef elm_jual* adr_jual;
9  typedef elm_sales* adr_sales;
10
11 struct sales{
12     string nama, gol;
13 };
14
15 struct elm_jual{
16     int info;
17     adr_jual next;
18 };
19
20 struct elm_sales{
21     sales info;
22     adr_sales next;
23     adr_jual nextJual;
24 };
25
26 struct mll{
27     adr_sales First;
28 };
29
30 void Create_list(mll &List_Sales){
31     List_Sales.First = NULL;
32 }
33
34 void new_elm_sales(sales info, adr_sales &S){
35     S = new elm_sales;
36     S->next = NULL;
37     S->nextJual = NULL;
38     S->info = info;
39 }

```

```

41 void new_elm_jual(int info, adr_jual &J){
42     J = new elm_jual;
43     J->info = info;
44 }
45
46 void insert_last_Sales(mll &List_Sales, adr_sales S){
47     if(List_Sales.First == NULL){
48         List_Sales.First = S;
49     }else{
50         adr_sales P = List_Sales.First;
51         while(P->next != NULL){
52             P = P->next;
53         }
54         P->next = S;
55     }
56 }
57
58 adr_sales Search_Sales(mll List_Sales, string nama_Sales){
59     adr_sales P = List_Sales.First;
60     while(P != NULL){
61         if(P->info.nama == nama_Sales){
62             return P;
63         }
64         P = P->next;
65     }
66     return NULL;
67 }

```

```

69 void Insert_new_Penjualan(mll &List_Sales, adr_sales S, adr_jual J){
70     if(S->nextJual == NULL){
71         S->nextJual = J;
72     }else{
73         adr_jual P = S->nextJual;
74         while(P->next != NULL){
75             P = P->next;
76         }
77         P->next = J;
78     }
79 }
80
81 void Delete_First_Penjualan(mll &List_Sales, adr_sales S, adr_jual &J){
82     J = S->nextJual;
83     S->nextJual = J->next;
84     J->next = NULL;
85 }
86
87 void Delete_after_Penjualan(mll &List_Sales, adr_sales S, adr_jual prec, adr_jual &J){
88     J = prec->next;
89     prec->next = J->next;
90     J->next = NULL;
91 }
92
93 void Delete_Last_Penjualan(mll &List_Sales, adr_sales S, adr_jual &J){
94     adr_jual P = S->nextJual;
95     adr_jual Prec = NULL;
96
97     while(P->next != NULL){
98         Prec = P;
99         P = P->next;
100    }
101
102    if(Prec == NULL){
103        S->nextJual = NULL;
104    }else{
105        Prec->next = NULL;
106    }
107
108    J = P;
109 }
110
111 void Show_data_sales(mll list_sales){
112     adr_sales P = list_sales.First;
113
114     while(P != NULL){
115         cout << P->info.nama << endl;
116
117         adr_jual Q = P->nextJual;
118
119         while(Q != NULL){
120             cout << Q->info << " ";
121             Q = Q->next;
122         }
123         cout << endl << endl;
124
125         P = P->next;
126     }
127 }
128

```

```

129 void Delete_Penjualan(mll &list_sales){
130     adr_sales P = list_sales.First;
131     adr_jual Q, prec, J;
132
133     while(P != NULL){
134         Q = P->nextJual;
135         prec = NULL;
136
137         while(Q != NULL){
138             if(Q->info < 3){
139                 if(Q == P->nextJual){
140                     Delete_First_Penjualan(list_sales, P, J);
141                     Q = P->nextJual;
142                 }else if(Q->next == NULL){
143                     Delete_Last_Penjualan(list_sales, P, J);
144                     Q = NULL;
145                 }else{
146                     Delete_after_Penjualan(list_sales, P, prec, J);
147                     Q = prec->next;
148                 }
149             }else{
150                 prec = Q;
151                 Q = Q->next;
152             }
153         }
154         P = P->next;
155     }
156 }

```

```

④ PS C:\shellyn\kuliah\semester3\strukturData> cd "c:\shellyn\kuliah\semester3\strukturData\teori\tugasMultiLinkedList\" ; if ($?) { g++ 1.cpp -o 1 } ; if ($?) { .\1 }
In file included from c:\mingw\include\stdio.h:69:0,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\cstdio:42,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\ext\string_versions.h:43,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\bits\basic_string.h:5402,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\string:52,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\bits\locale_classemes.h:40,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\bits\ios_base.h:41,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\ios:42,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\ostream:38,
                 from c:\mingw\lib\gcc\mingw32\6.3.0\include\c++\iostream:39,
                 from 1.cpp:1:
1.cpp: In function 'void Show_data_sales(mll)':
1.cpp:119:20: error: cannot convert 'bool' to 'adr_jual {aka elm_jual*}' in assignment
           while(Q != NULL){
                           ^

```

```

158 int main(){
159     mll showroom;
160     Create_list(showroom);
161
162     adr_sales s1, s2, s3, s4, s5;
163
164     new_elm_sales({"Siska", "A"}, s1);
165     new_elm_sales({"Affan", "C"}, s2);
166     new_elm_sales({"Cahya", "A"}, s3);
167     new_elm_sales("Lia", "C"), s4);
168     new_elm_sales("Hafidz", "B"}, s5);
169
170     insert_last_Sales(showroom, s1);
171     insert_last_Sales(showroom, s2);
172     insert_last_Sales(showroom, s3);
173     insert_last_Sales(showroom, s4);
174     insert_last_Sales(showroom, s5);
175
176
177     adr_jual j;
178
179     new_elm_jual(5, j); Insert_new_Penjualan(showroom, s1, j);
180     new_elm_jual(3, j); Insert_new_Penjualan(showroom, s1, j);
181     new_elm_jual(4, j); Insert_new_Penjualan(showroom, s1, j);
182     new_elm_jual(2, j); Insert_new_Penjualan(showroom, s1, j);
183
184     new_elm_jual(4, j); Insert_new_Penjualan(showroom, s2, j);
185     new_elm_jual(3, j); Insert_new_Penjualan(showroom, s2, j);
186
187     new_elm_jual(5, j); Insert_new_Penjualan(showroom, s3, j);
188     new_elm_jual(0, j); Insert_new_Penjualan(showroom, s3, j);
189     new_elm_jual(2, j); Insert_new_Penjualan(showroom, s3, j);
190
191     new_elm_jual(7, j); Insert_new_Penjualan(showroom, s5, j);
192
193     Show_data_sales(showroom);
194
195     return 0;
196 }

```

- PS C:\shellyn\kuliah\semester3\strukturData\teori\tugasMultiLinkedList> cd "c:\shellyn\kuliah\semester3\strukturData\teori\tugasMultiLinkedList\" ; if (\$?) { g++ 2.cpp -o 2 } ; if (\$?) { .\2 }

Siska

5 3 4 2

Affan

4 3

Cahya

5 0 2

Lia

Hafidz

7

```

1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct elm_jual;
6 struct elm_sales;
7
8 typedef elm_jual* adr_jual;
9 typedef elm_sales* adr_sales;
10
11 struct sales{
12     string nama, gol;
13 };
14
15 struct elm_jual{
16     int info;
17     adr_jual next;
18 };
19
20 struct elm_sales{
21     sales info;
22     adr_sales next;
23     adr_jual nextJual;
24 };
25
26 struct mll{
27     adr_sales First;
28 };
29
30 void Create_list(mll &List_Sales){
31     List_Sales.First = NULL;
32 }
33
34 void new_elm_sales(sales info, adr_sales &S){
35     S = new elm_sales;
36     S->next = NULL;
37     S->nextJual = NULL;
38     S->info = info;
39 }
40
41 void new_elm_jual(int info, adr_jual &J){
42     J = new elm_jual;
43     J->info = info;
44     J->next = NULL;
45 }
46
47 void insert_last_Sales(mll &List_Sales, adr_sales S){
48     if(List_Sales.First == NULL){
49         List_Sales.First = S;
50     }else{
51         adr_sales P = List_Sales.First;
52         while(P->next != NULL){
53             P = P->next;
54         }
55         P->next = S;
56     }
57 }
58
59 adr_sales Search_Sales(mll List_Sales, string nama_Sales){
60     adr_sales P = List_Sales.First;
61     while(P != NULL){
62         if(P->info.nama == nama_Sales){
63             return P;
64         }
65         P = P->next;
66     }
67     return NULL;
68 }
```

```

70 void Insert_new_Penjualan(mll &List_Sales, adr_sales S, adr_jual J){
71     if(S->nextJual == NULL){
72         S->nextJual = J;
73     }else{
74         adr_jual P = S->nextJual;
75         while(P->next != NULL){
76             P = P->next;
77         }
78         P->next = J;
79     }
80 }
81
82 void Delete_First_Penjualan(mll &List_Sales, adr_sales S, adr_jual &J){
83     J = S->nextJual;
84     S->nextJual = J->next;
85     J->next = NULL;
86 }
87
88 void Delete_after_Penjualan(mll &List_Sales, adr_sales S, adr_jual prec, adr_jual &J){
89     J = prec->next;
90     prec->next = J->next;
91     J->next = NULL;
92 }
93
94 void Delete_Last_Penjualan(mll &List_Sales, adr_sales S, adr_jual &J){
95     adr_jual P = S->nextJual;
96     adr_jual Prec = NULL;
97
98     while(P->next != NULL){
99         Prec = P;
100        P = P->next;
101    }
102
103    if(Prec == NULL){
104        S->nextJual = NULL;
105    }else{
106        Prec->next = NULL;
107    }
108
109    J = P;
110 }
111
112 void Show_data_sales(mll list_sales){
113     adr_sales P = list_sales.First;
114
115     while(P != NULL){
116         cout << P->info.nama << endl;
117
118         adr_jual Q = P->nextJual;
119
120         while(Q != NULL){
121             cout << Q->info << " ";
122             Q = Q->next;
123         }
124         cout << endl << endl;
125
126         P = P->next;
127     }
128 }
129

```

```

130 void Delete_Penjualan(mll &list_sales){
131     adr_sales P = list_sales.First;
132     adr_jual Q, prec, J;
133
134     while(P != NULL){
135         Q = P->nextJual;
136         prec = NULL;
137
138         while(Q != NULL){
139             if(Q->info < 3){
140                 if(Q == P->nextJual){
141                     Delete_First_Penjualan(list_sales, P, J);
142                     Q = P->nextJual;
143                 }else if(Q->next == NULL){
144                     Delete_Last_Penjualan(list_sales, P, J);
145                     Q = NULL;
146                 }else{
147                     Delete_after_Penjualan(list_sales, P, prec, J);
148                     Q = prec->next;
149                 }
150             }else{
151                 prec = Q;
152                 Q = Q->next;
153             }
154         }
155         P = P->next;
156     }
157 }
158
159 int main(){
160     mll showroom;
161     Create_list(showroom);
162
163     adr_sales s1, s2, s3, s4, s5;
164
165     new_elm_sales({"Siska", "A"}, s1);
166     new_elm_sales({"Affan", "C"}, s2);
167     new_elm_sales({"Cahya", "A"}, s3);
168     new_elm_sales({"Lia", "C"}, s4);
169     new_elm_sales({"Hafidz", "B"}, s5);
170
171     insert_last_Sales(showroom, s1);
172     insert_last_Sales(showroom, s2);
173     insert_last_Sales(showroom, s3);
174     insert_last_Sales(showroom, s4);
175     insert_last_Sales(showroom, s5);
176
177     adr_jual j;
178
179     new_elm_jual(5, j); Insert_new_Penjualan(showroom, s1, j);
180     new_elm_jual(3, j); Insert_new_Penjualan(showroom, s1, j);
181     new_elm_jual(4, j); Insert_new_Penjualan(showroom, s1, j);
182     new_elm_jual(2, j); Insert_new_Penjualan(showroom, s1, j);
183
184     new_elm_jual(4, j); Insert_new_Penjualan(showroom, s2, j);
185     new_elm_jual(3, j); Insert_new_Penjualan(showroom, s2, j);
186
187     new_elm_jual(5, j); Insert_new_Penjualan(showroom, s3, j);
188     new_elm_jual(0, j); Insert_new_Penjualan(showroom, s3, j);
189     new_elm_jual(2, j); Insert_new_Penjualan(showroom, s3, j);
190
191     new_elm_jual(7, j); Insert_new_Penjualan(showroom, s5, j);
192
193     Show_data_sales(showroom);
194
195     return 0;
196 }
```