

Nama : Helmi Efendi Lubis  
NIM : 1301223338  
Kelas : IF-46-08

## Tugas Pendahuluan Modul 8

1.

doubleCircular.h

```
doubleCircular.h
1  #ifndef DOUBLECIRCULAR_H_INCLUDED
2  #define DOUBLECIRCULAR_H_INCLUDED
3
4  #include <iostream>
5  using namespace std;
6
7  typedef int infotype;
8  typedef struct ElmList *address;
9  struct ElmList {
10     infotype info;
11     address next;
12 };
13 struct List {
14     address first;
15     address last;
16 };
17
18 void createList_1301223338(List &L);
19 address alokasi_1301223338(infotype x);
20 void dealokasi_1301223338(address &P);
21 void insertFirst_1301223338(List &L, address P);
22 void insertAfter_1301223338(List &L, address Prec, address P);
23 void insertLast_1301223338(List &L, address P);
24 address findElm_1301223338(List L, infotype x);
25 void deleteFirst_1301223338(List &L, address &P);
26 void deleteAfter_1301223338(List &L, address Prec, address &P);
27 void deleteLast_1301223338(List &L, address &P);
28 void printInfo_1301223338(List L);
29
30 void deleteByValue_1301223338(List &L, infotype x);
31 void reverseList_1301223338(List L, List &L2);
32
33 #endif // DOUBLECIRCULAR_H_INCLUDED
34
```

doubleCircular.cpp

## doubleCircular.cpp

```
1  #include "doubleCircular.h"
2
3  void createList_1301223338(List &L){
4      L.first = NULL;
5      L.last = NULL;
6  }
7
8  address alokasi_1301223338(infotype x) {
9      address p = new ElmList;
10     p->info = x;
11     p->next = NULL;
12     return p;
13 }
14
15 void dealokasi_1301223338(address &P){
16     P = NULL;
17 }
18
19 void insertFirst_1301223338(List &L, address P){
20     // address q;
21     if (L.first == NULL)
22     {
23         L.first = P;
24         L.last = P;
25         P->next = L.first;
26     }
27     else
28     {
29         // q = L.first;
30         // while(q->next != L.first){
31         //     q = q->next;
32         // }
33         P->next = L.first;
34         L.last->next = P;
35         L.first = P;
36     }
37 }
38
```

## doubleCircular.cpp

```
38
39 void insertAfter_1301223338(List &L, address Prec, address P){
40     if(Prec->next == L.first){
41         insertLast_1301223338(L, Prec);
42     }else {
43         P->next = Prec->next;
44         Prec->next = P;
45     }
46 }
47
48 void insertLast_1301223338(List &L, address P) {
49     if (L.first == NULL)
50     {
51         L.first = P;
52         L.last = P;
53         P->next = L.first;
54     }
55     else
56     {
57         P->next = L.first;
58         L.last->next = P;
59         L.last = P;
60     }
61 }
62
```

## doubleCircular.cpp

```
62 |
63 | address findElm_1301223338(List L, infotype x) {
64 |     address p;
65 |     if (L.first == NULL)
66 |     {
67 |         return NULL;
68 |     }
69 |     else
70 |     {
71 |         p = L.first;
72 |         while (p->next != L.first)
73 |         {
74 |             if(p->info == x){
75 |                 return p;
76 |             }
77 |             p = p->next;
78 |         }
79 |         if(p->info == x){
80 |             return p;
81 |         }
82 |     }
83 |     return NULL;
84 | }
85 |
86 | void deleteFirst_1301223338(List &L, address &P) {
87 |     if(L.first == NULL){
88 |         P = NULL;
89 |     }else if(L.first->next == L.first){
90 |         P = L.first;
91 |         P->next = NULL;
92 |         L.first = NULL;
93 |         L.last = NULL;
94 |     }
95 |     else {
96 |         P = L.first;
97 |         L.first = P->next;
98 |         P->next = NULL;
99 |         L.last->next = L.first;
100 |     }
101 | }
102 |
```

```
doubleCircular.cpp x
102
103 void deleteAfter_1301223338(List &L, address Prec, address &P){
104     if(L.first == NULL){
105         P = NULL;
106     }else if(Prec->next == L.first){
107         deleteFirst_1301223338(L, P);
108     }else {
109         P = Prec->next;
110         Prec->next = P->next;
111         P->next = NULL;
112     }
113 }
114
115 void deleteLast_1301223338(List &L, address &P) {
116     address q;
117     if (L.first == NULL)
118     {
119         P = NULL;
120     }else if(L.first->next == L.first){
121         P = L.first;
122         P->next = NULL;
123         L.first = NULL;
124         L.last = NULL;
125     }
126     else
127     {
128         q = L.first;
129         while(q->next->next != L.first){
130             q = q->next;
131         }
132         P = L.last;
133         L.last = q;
134         P->next = NULL;
135         L.last->next = L.first;
136     }
137 }
138
```

```
doubleCircular.cpp x
138
139 void printInfo_1301223338(List L){
140     address p = L.first;
141     if(L.first == NULL){
142         cout << "List Kosong" << endl;
143     }else {
144         while(p->next != L.first){
145             cout << p->info << ", ";
146             p = p->next;
147         }
148         cout << p->info << endl;
149     }
150 }
151
```

2.  
doubleCircular.cpp

```
doubleCircular.cpp
151
152 void deleteByValue_1301223338(List &L, infotype x){
153     address p,q,temp;
154     if (L.first == NULL)
155     {
156         cout << "List Kosong" << endl;
157     }
158     else
159     {
160         p = findElm_1301223338(L, x);
161         if(p != NULL){
162             if(p == L.first){
163                 deleteFirst_1301223338(L, temp);
164             }else if(p == L.last){
165                 deleteLast_1301223338(L, temp);
166             }else {
167                 q = L.first;
168                 while(q->next != p){
169                     q = q->next;
170                 }
171                 deleteAfter_1301223338(L, q, temp);
172             }
173         }
174     }
175 }
176

176
177 void reverseList_1301223338(List L, List &L2){
178     address p,temp;
179     if (L.first == NULL)
180     {
181         cout << "List Kosong" << endl;
182     }
183     else
184     {
185         p = L.first;
186         while(p->next != L.first){
187             deleteFirst_1301223338(L, temp);
188             insertFirst_1301223338(L2, temp);
189             p = L.first;
190         }
191         if(p->next == L.first){
192             deleteFirst_1301223338(L, temp);
193             insertFirst_1301223338(L2, temp);
194         }
195     }
196 }
197
```

Main.cpp

```
main.cpp x
1  #include <iostream>
2  #include "doubleCircular.h"
3  #include "doubleCircular.cpp"
4
5  using namespace std;
6
7  int main()
8  {
9      List L, L2;
10     createList_1301223338(L);
11     insertLast_1301223338(L, alokasi_1301223338(5));
12     insertLast_1301223338(L, alokasi_1301223338(15));
13     insertLast_1301223338(L, alokasi_1301223338(25));
14     insertLast_1301223338(L, alokasi_1301223338(35));
15     insertLast_1301223338(L, alokasi_1301223338(45));
16     insertLast_1301223338(L, alokasi_1301223338(55));
17     insertLast_1301223338(L, alokasi_1301223338(65));
18     insertLast_1301223338(L, alokasi_1301223338(75));
19     insertLast_1301223338(L, alokasi_1301223338(85));
20     cout << "Isi List Awal: " << endl;
21     printInfo_1301223338(L);
22
23     // cout << "Isi List reverse: " << endl;
24     // reverseList_1301223338(L, L2);
25     // printInfo_1301223338(L2);
26
27     deleteByValue_1301223338(L, 5);
28     deleteByValue_1301223338(L, 100);
29     deleteByValue_1301223338(L, 45);
30     deleteByValue_1301223338(L, 85);
31     cout << "Isi List Setelah dihapus: " << endl;
32     printInfo_1301223338(L);
33 }
34
```

## Output

```
mac — test — 80x24
Last login: Sat Nov  4 18:35:37 on ttys014
mac@192 ~ % /Users/mac/Documents/COOLYEAH/KULIAH\ SEMESTER\ 3/Struktur\ Data/Pra
ktikum/TP\ MODUL\ 8/Mod8/test ; exit;
Isi List Awal:
5, 15, 25, 35, 45, 55, 65, 75, 85
Isi List Setelah dihapus:
15, 25, 35, 55, 65, 75

[Process completed]
```