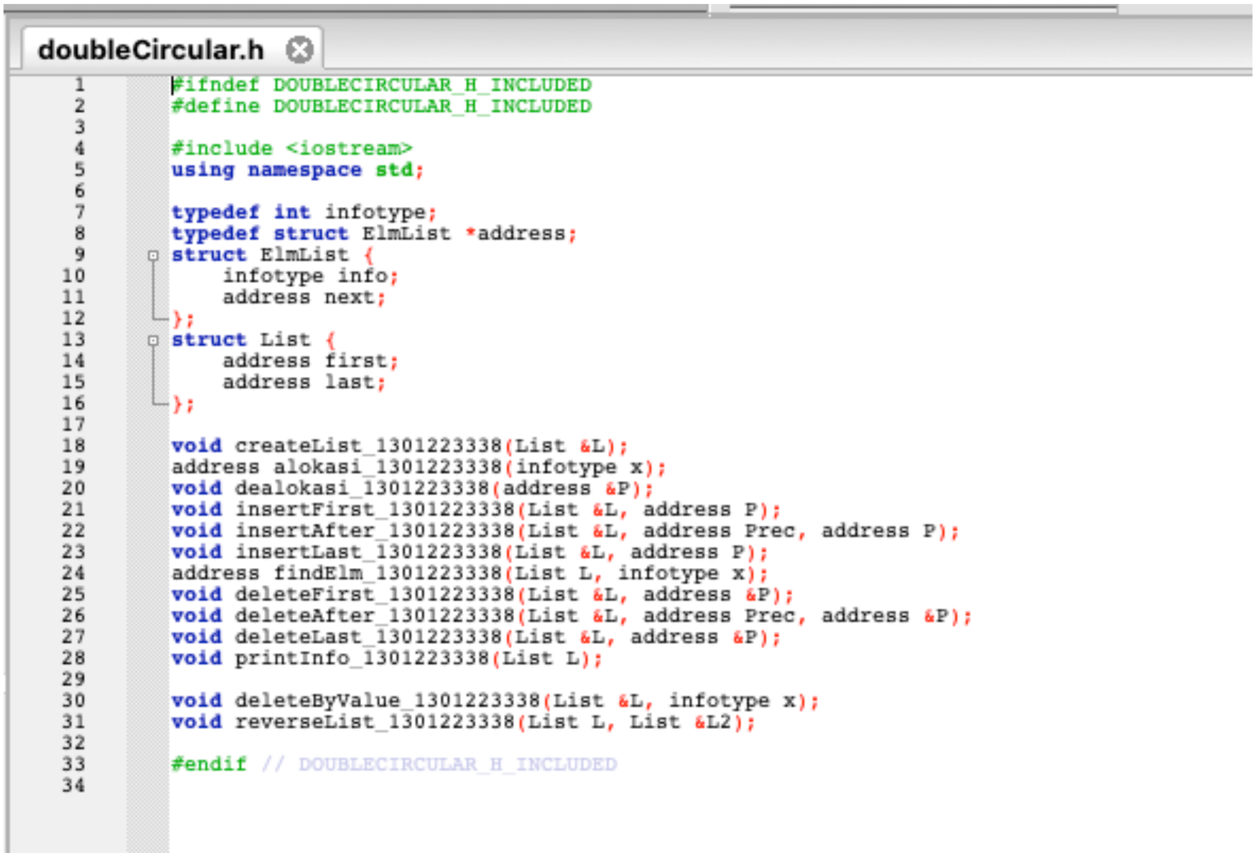
Nama   : Helmi Efendi Lubis
NIM   : 1301223338
Kelas   : IF-46-08

Tugas Pendahuluan Modul 8

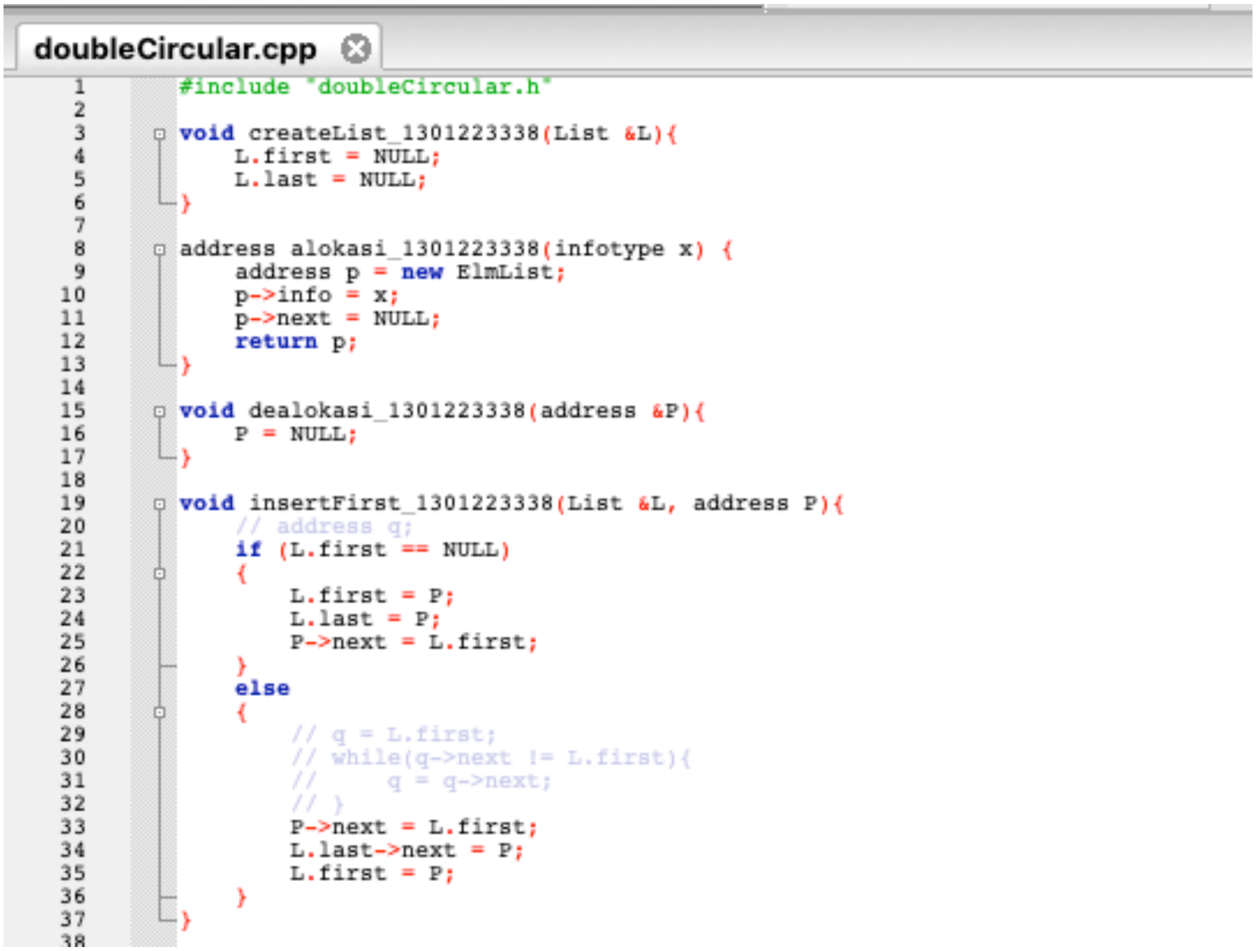
1.

doubleCircular.h

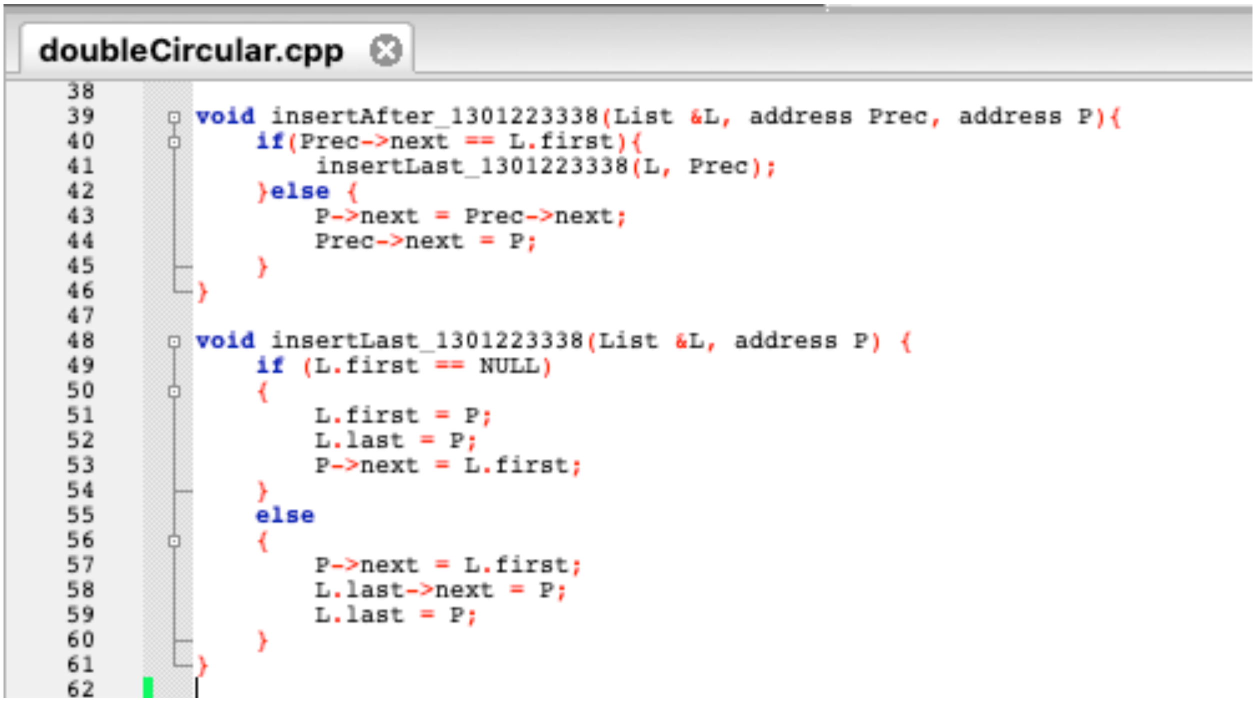


```
#ifndef DOUBLECIRCULAR_H_INCLUDED
#define DOUBLECIRCULAR_H_INCLUDED

#include <iostream>
using namespace std;

typedef int infotype;
typedef struct ElmList *address;
struct ElmList {
    infotype info;
    address next;
};
struct List {
    address first;
    address last;
};

void createList_1301223338(List &L);
address alokasi_1301223338(infotype x);
void dealokasi_1301223338(address &P);
void insertFirst_1301223338(List &L, address P);
void insertAfter_1301223338(List &L, address Prec, address P);
void insertLast_1301223338(List &L, address P);
address findElm_1301223338(List L, infotype x);
void deleteFirst_1301223338(List &L, address &P);
void deleteAfter_1301223338(List &L, address Prec, address &P);
void deleteLast_1301223338(List &L, address &P);
void printInfo_1301223338(List L);

void deleteByValue_1301223338(List &L, infotype x);
void reverseList_1301223338(List L, List &L2);

#endif // DOUBLECIRCULAR_H_INCLUDED
```

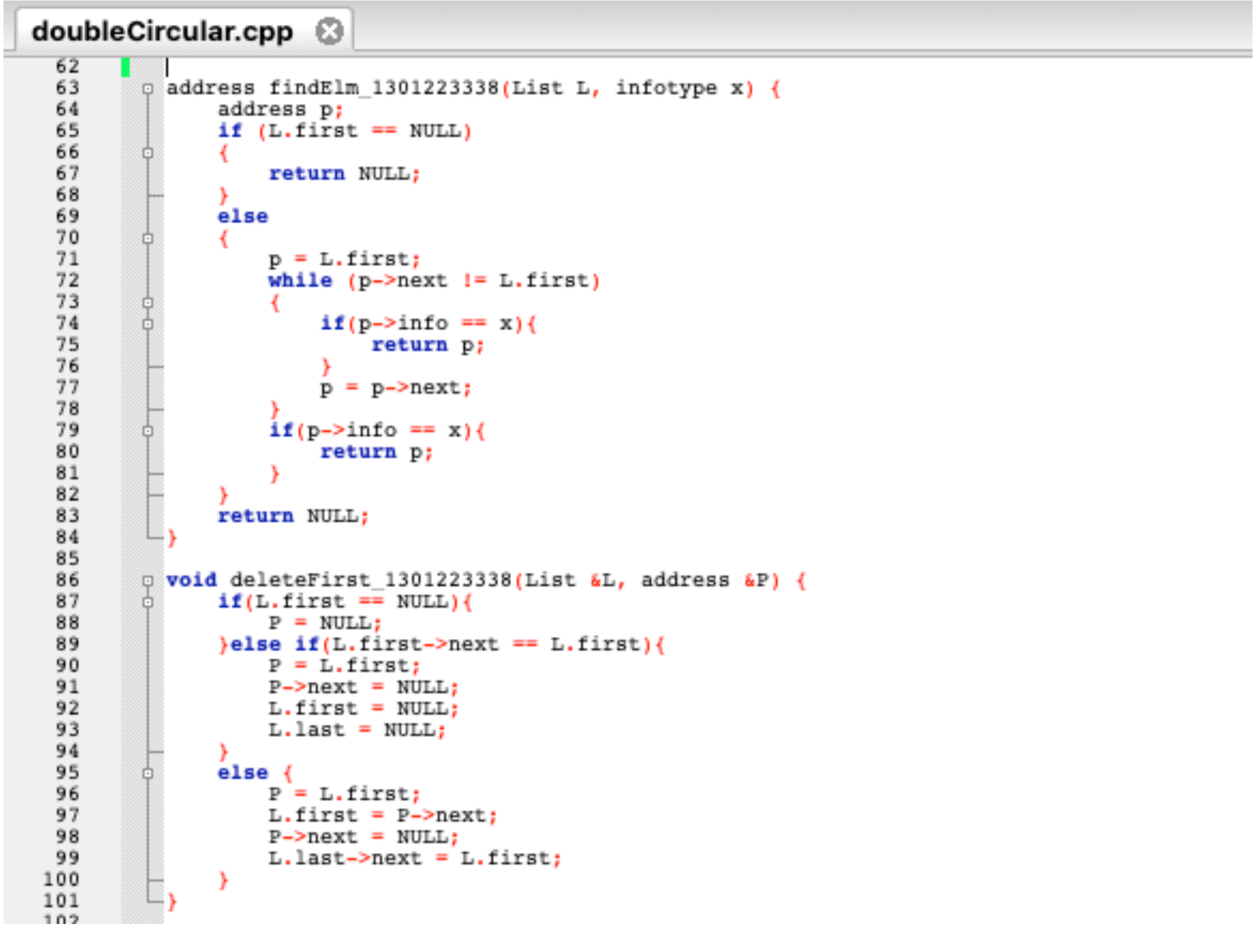
doubleCircular.cpp



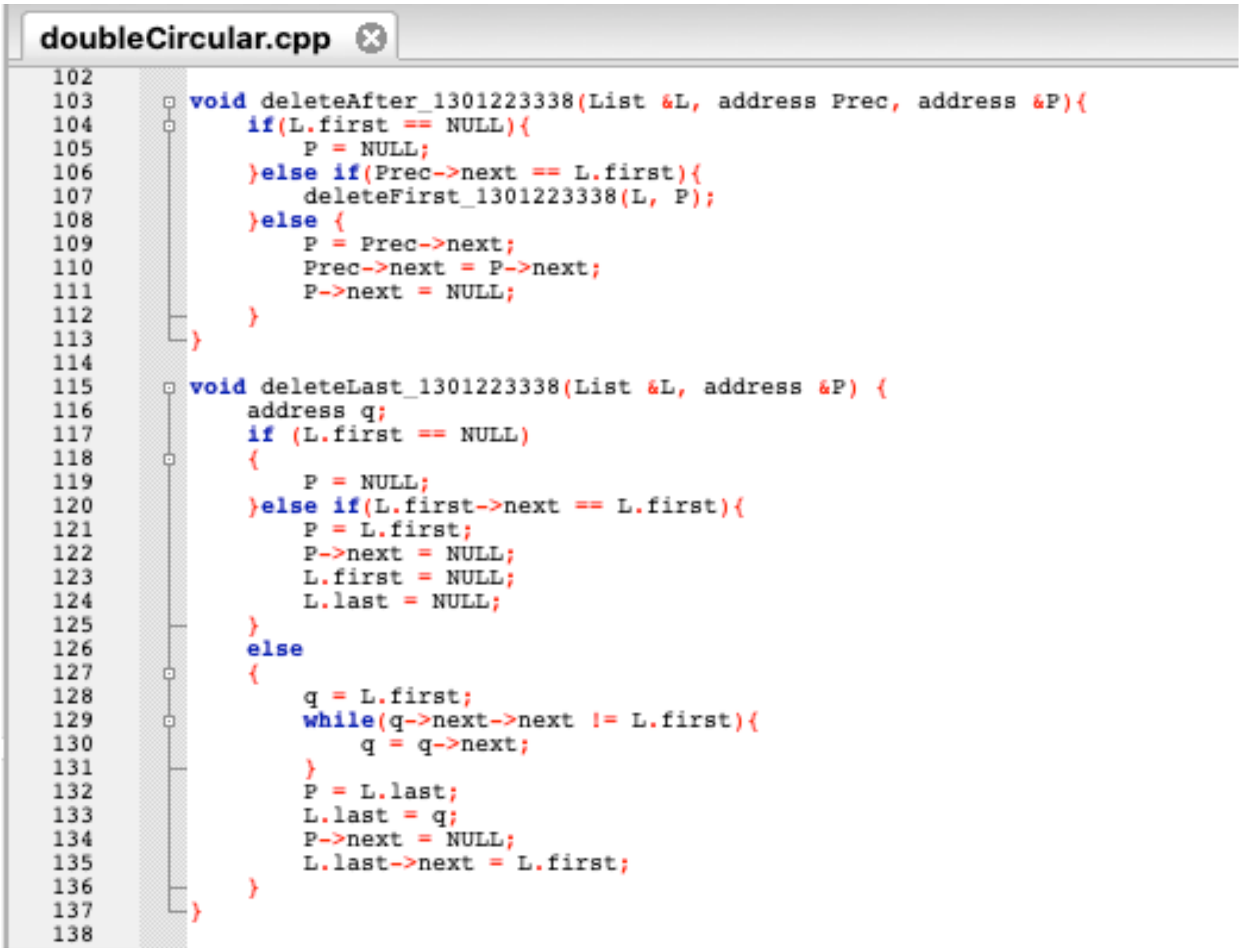
```cpp
#include "doubleCircular.h"

void createList_1301223338(List &L){
    L.first = NULL;
    L.last = NULL;
}

address alokasi_1301223338(infotype x) {
    address p = new ElmList;
    p->info = x;
    p->next = NULL;
    return p;
}

void dealokasi_1301223338(address &P){
    P = NULL;
}

void insertFirst_1301223338(List &L, address P){
    // address q;
    if (L.first == NULL)
    {
        L.first = P;
        L.last = P;
        P->next = L.first;
    }
    else
    {
        // q = L.first;
        // while(q->next != L.first){
        //      q = q->next;
        // }
        P->next = L.first;
        L.last->next = P;
        L.first = P;
    }
}

```



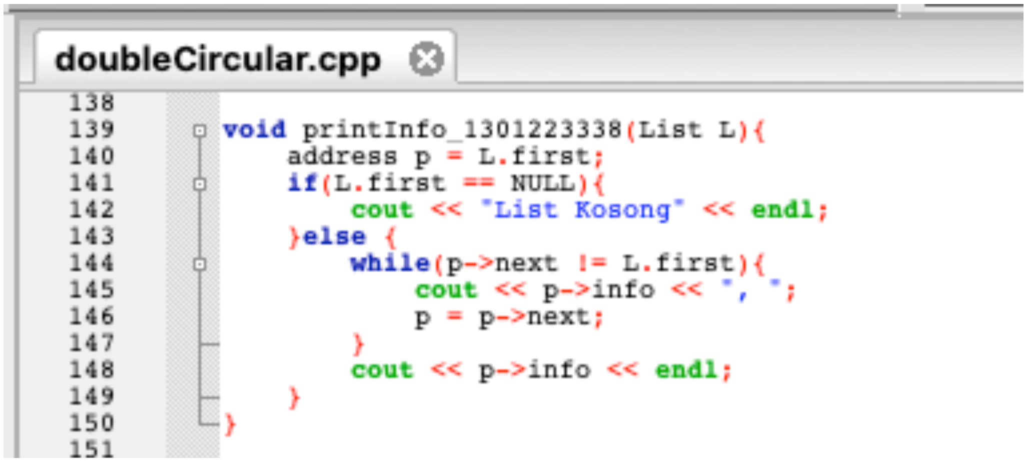
```cpp

void insertAfter_1301223338(List &L, address Prec, address P){
    if(Prec->next == L.first){
        insertLast_1301223338(L, Prec);
    }else {
        P->next = Prec->next;
        Prec->next = P;
    }
}

void insertLast_1301223338(List &L, address P) {
    if (L.first == NULL)
    {
        L.first = P;
        L.last = P;
        P->next = L.first;
    }
    else
    {
        P->next = L.first;
        L.last->next = P;
        L.last = P;
    }
}

```



```cpp

address findElm_1301223338(List L, infotype x) {
    address p;
    if (L.first == NULL)
    {
        return NULL;
    }
    else
    {
        p = L.first;
        while (p->next != L.first)
        {
            if(p->info == x){
                return p;
            }
            p = p->next;
        }
        if(p->info == x){
            return p;
        }
    }
    return NULL;
}

void deleteFirst_1301223338(List &L, address &P) {
    if(L.first == NULL){
        P = NULL;
    }else if(L.first->next == L.first){
        P = L.first;
        P->next = NULL;
        L.first = NULL;
        L.last = NULL;
    }
    else {
        P = L.first;
        L.first = P->next;
        P->next = NULL;
        L.last->next = L.first;
    }
}
```

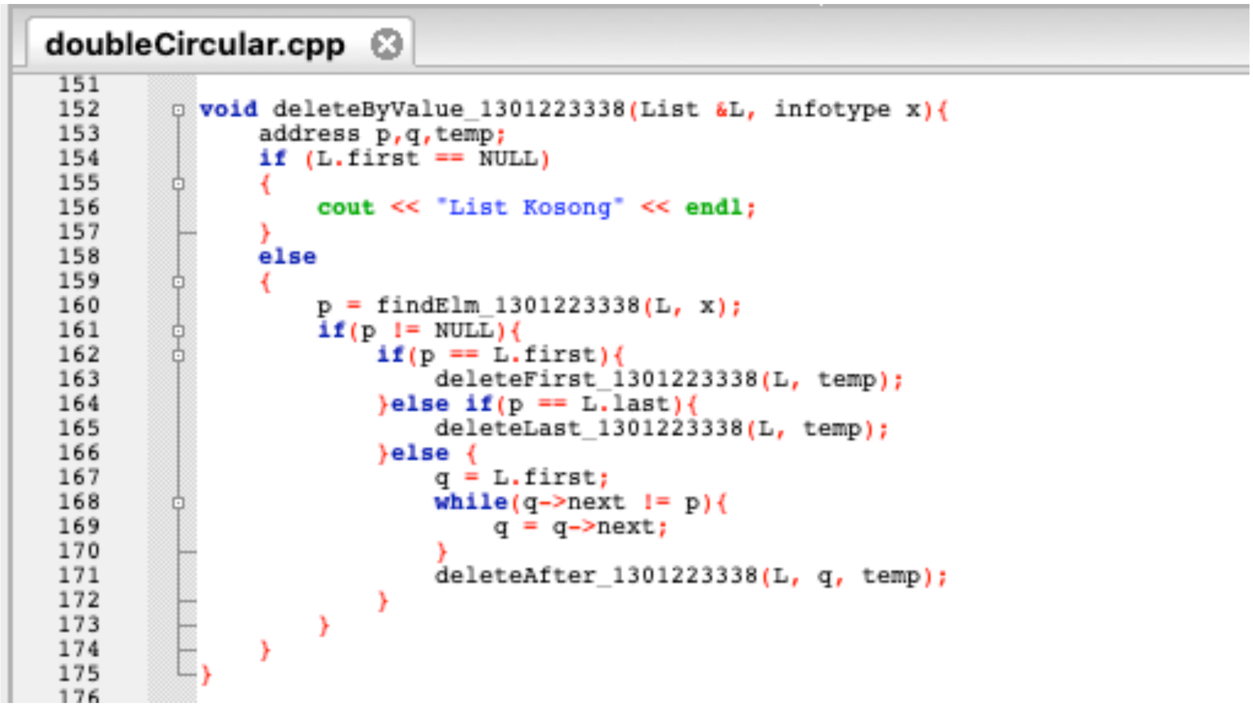


```
void deleteAfter_1301223338(List &L, address Prec, address &P){
    if(L.first == NULL){
        P = NULL;
    }else if(Prec->next == L.first){
        deleteFirst_1301223338(L, P);
    }else {
        P = Prec->next;
        Prec->next = P->next;
        P->next = NULL;
    }
}

void deleteLast_1301223338(List &L, address &P) {
    address q;
    if (L.first == NULL)
    {
        P = NULL;
    }else if(L.first->next == L.first){
        P = L.first;
        P->next = NULL;
        L.first = NULL;
        L.last = NULL;
    }
    else
    {
        q = L.first;
        while(q->next->next != L.first){
            q = q->next;
        }
        P = L.last;
        L.last = q;
        P->next = NULL;
        L.last->next = L.first;
    }
}
```

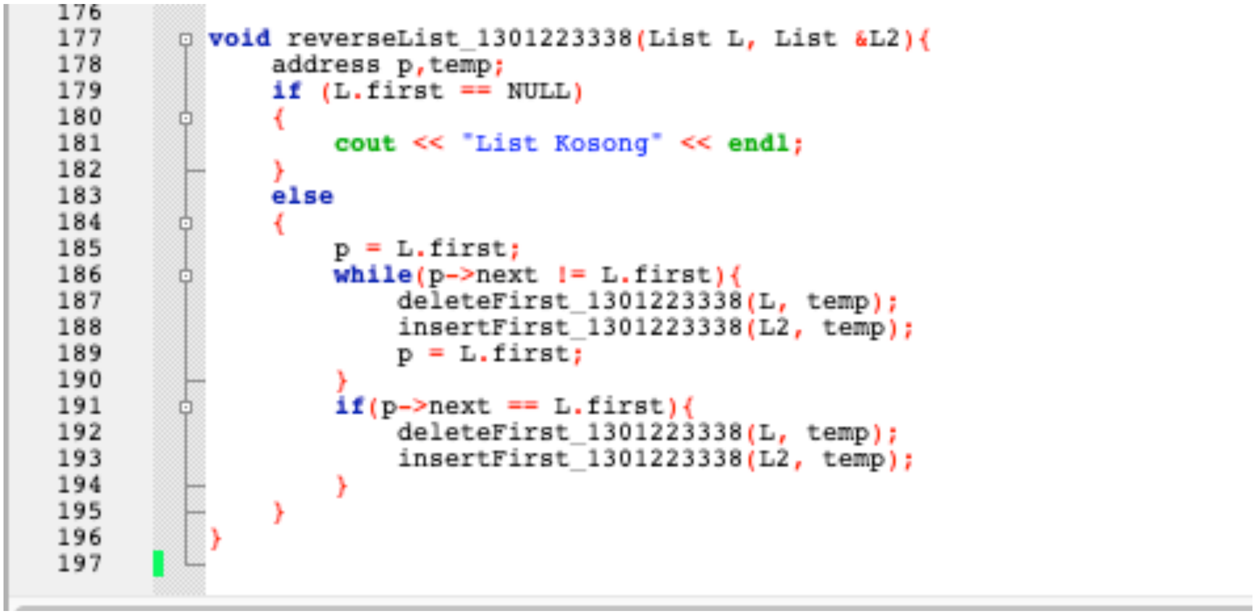


```
void printInfo_1301223338(List L){
    address p = L.first;
    if(L.first == NULL){
        cout << "List Kosong" << endl;
    }else {
        while(p->next != L.first){
            cout << p->info << ", ";
            p = p->next;
        }
        cout << p->info << endl;
    }
}
```

2.
doubleCircular.cpp

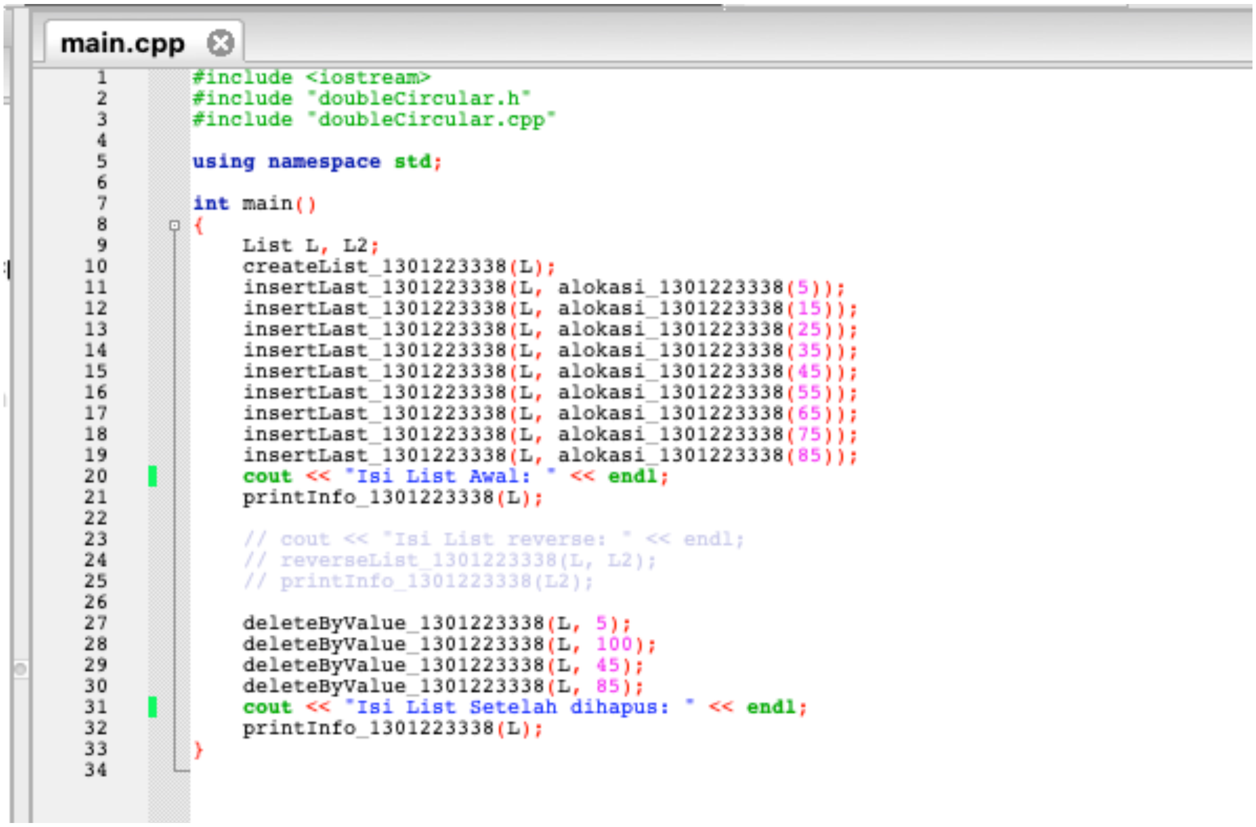


```cpp

void deleteByValue_1301223338(List &L, infotype x){
    address p,q,temp;
    if (L.first == NULL)
    {
        cout << "List Kosong" << endl;
    }
    else
    {
        p = findElm_1301223338(L, x);
        if(p != NULL){
            if(p == L.first){
                deleteFirst_1301223338(L, temp);
            }else if(p == L.last){
                deleteLast_1301223338(L, temp);
            }else {
                q = L.first;
                while(q->next != p){
                    q = q->next;
                }
                deleteAfter_1301223338(L, q, temp);
            }
        }
    }
}
```



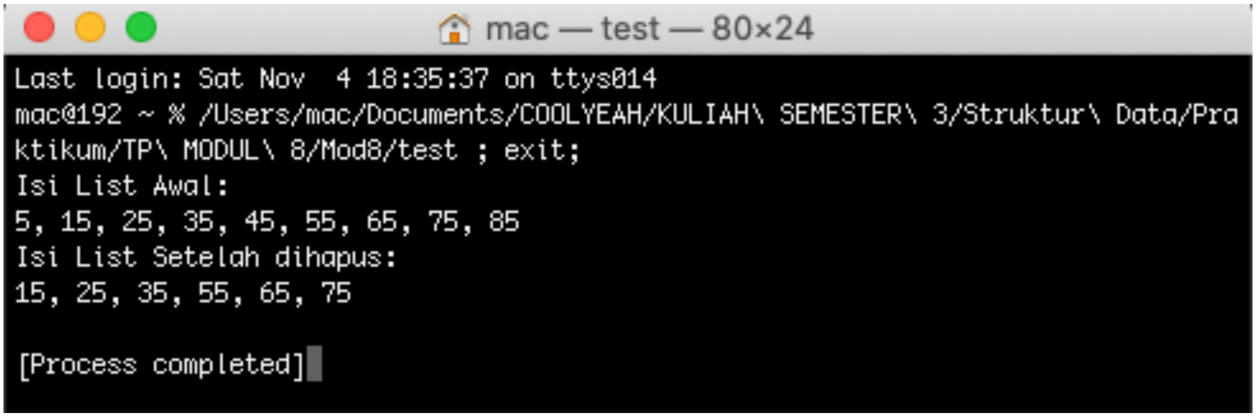
```cpp

void reverseList_1301223338(List L, List &L2){
    address p,temp;
    if (L.first == NULL)
    {
        cout << "List Kosong" << endl;
    }
    else
    {
        p = L.first;
        while(p->next != L.first){
            deleteFirst_1301223338(L, temp);
            insertFirst_1301223338(L2, temp);
            p = L.first;
        }
        if(p->next == L.first){
            deleteFirst_1301223338(L, temp);
            insertFirst_1301223338(L2, temp);
        }
    }
}

```

Main.cpp



```cpp
#include <iostream>
#include "doubleCircular.h"
#include "doubleCircular.cpp"

using namespace std;

int main()
{
    List L, L2;
    createList_1301223338(L);
    insertLast_1301223338(L, alokasi_1301223338(5));
    insertLast_1301223338(L, alokasi_1301223338(15));
    insertLast_1301223338(L, alokasi_1301223338(25));
    insertLast_1301223338(L, alokasi_1301223338(35));
    insertLast_1301223338(L, alokasi_1301223338(45));
    insertLast_1301223338(L, alokasi_1301223338(55));
    insertLast_1301223338(L, alokasi_1301223338(65));
    insertLast_1301223338(L, alokasi_1301223338(75));
    insertLast_1301223338(L, alokasi_1301223338(85));
    cout << "Isi List Awal: " << endl;
    printInfo_1301223338(L);

    // cout << "Isi List reverse: " << endl;
    // reverseList_1301223338(L, L2);
    // printInfo_1301223338(L2);

    deleteByValue_1301223338(L, 5);
    deleteByValue_1301223338(L, 100);
    deleteByValue_1301223338(L, 45);
    deleteByValue_1301223338(L, 85);
    cout << "Isi List Setelah dihapus: " << endl;
    printInfo_1301223338(L);
}
```

Output



```
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]
```