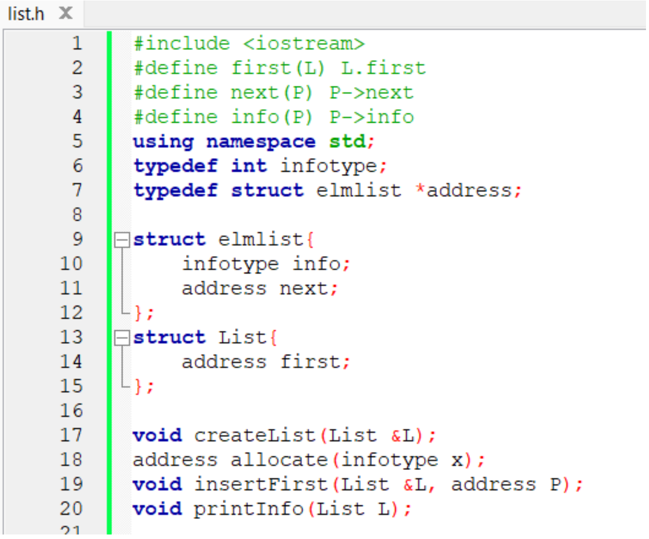
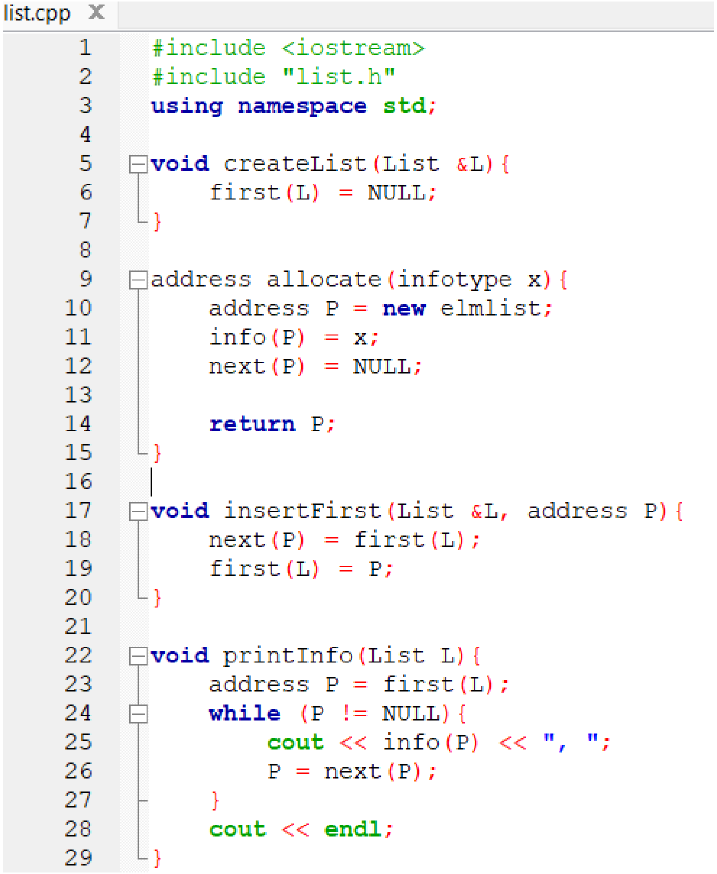
TP Modul 4 - Single Linked List (part 1)

- list.h :



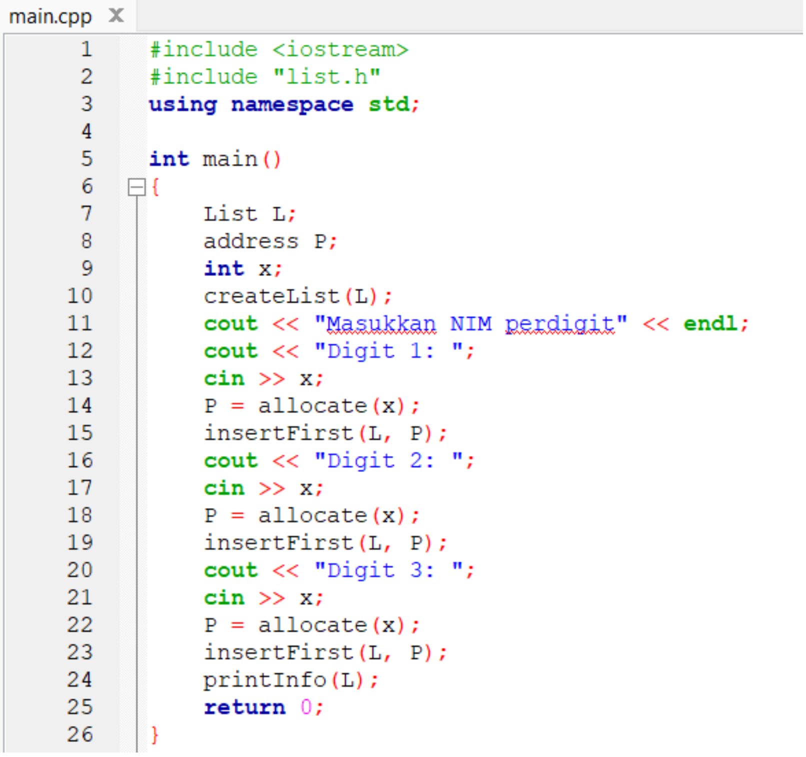
```cpp
#include <iostream>
#define first(L) L.first
#define next(P) P->next
#define info(P) P->info
using namespace std;
typedef int infotype;
typedef struct elmlist *address;

struct elmlist{
    infotype info;
    address next;
};
struct List{
    address first;
};

void createList(List &L);
address allocate(infotype x);
void insertFirst(List &L, address P);
void printInfo(List L);
```

- list.cpp :



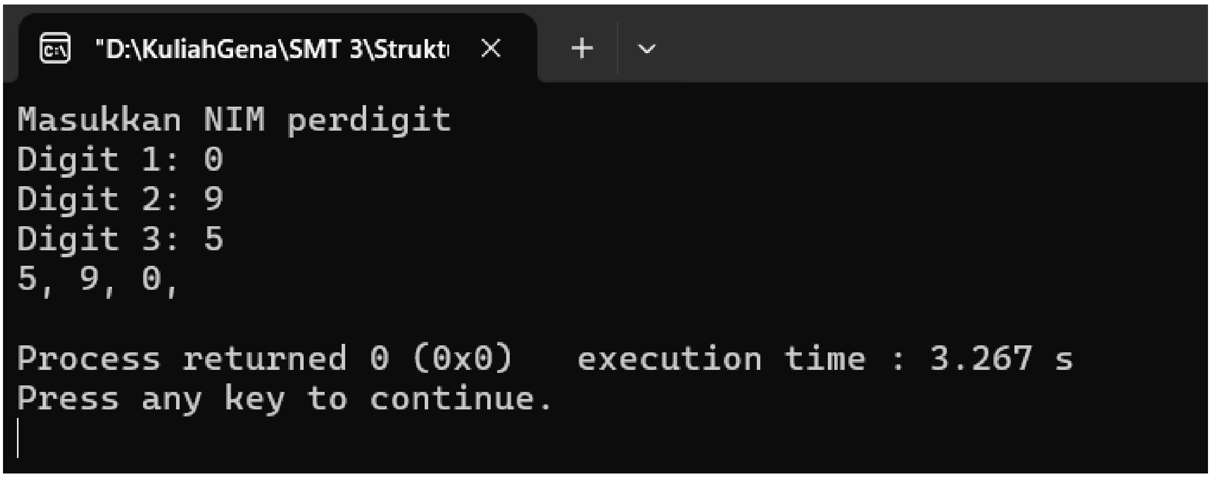
```cpp
#include <iostream>
#include "list.h"
using namespace std;

void createList(List &L){
    first(L) = NULL;
}

address allocate(infotype x){
    address P = new elmlist;
    info(P) = x;
    next(P) = NULL;

    return P;
}

void insertFirst(List &L, address P){
    next(P) = first(L);
    first(L) = P;
}

void printInfo(List L){
    address P = first(L);
    while (P != NULL){
        cout << info(P) << ", ";
        P = next(P);
    }
    cout << endl;
}
```

- main.cpp :



```cpp
#include <iostream>
#include "list.h"
using namespace std;

int main()
{
    List L;
    address P;
    int x;
    createList(L);
    cout << "Masukkan NIM perdigit" << endl;
    cout << "Digit 1: ";
    cin >> x;
    P = allocate(x);
    insertFirst(L, P);
    cout << "Digit 2: ";
    cin >> x;
    P = allocate(x);
    insertFirst(L, P);
    cout << "Digit 3: ";
    cin >> x;
    P = allocate(x);
    insertFirst(L, P);
    printInfo(L);
    return 0;
}
```

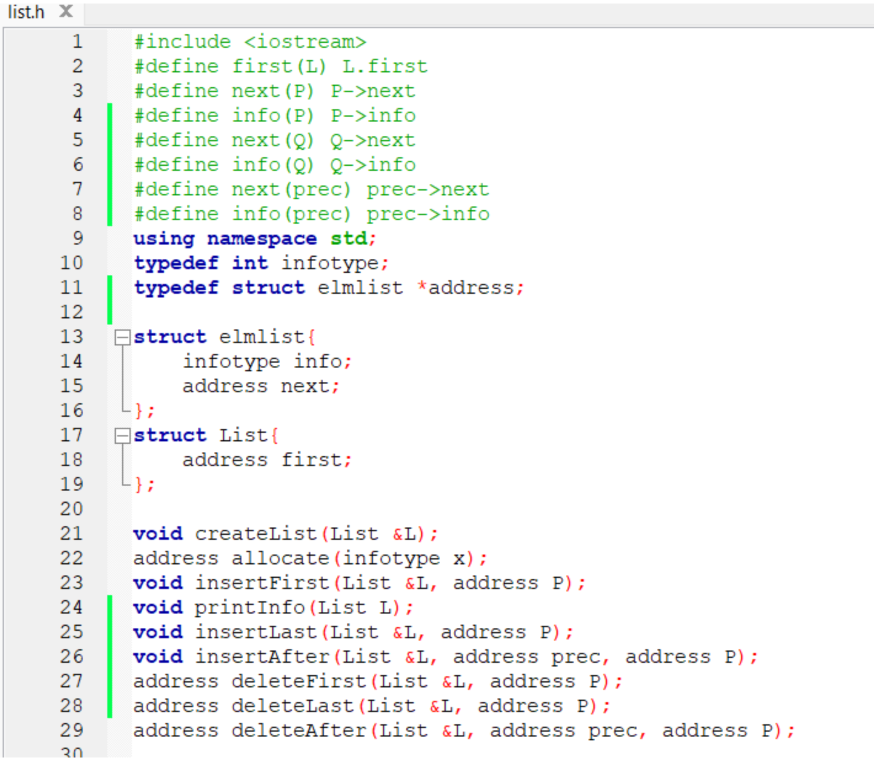
- output :



```
Masukkan NIM perdigit
Digit 1: 0
Digit 2: 9
Digit 3: 5
5, 9, 0,

Process returned 0 (0x0)   execution time : 3.267 s
Press any key to continue.
```

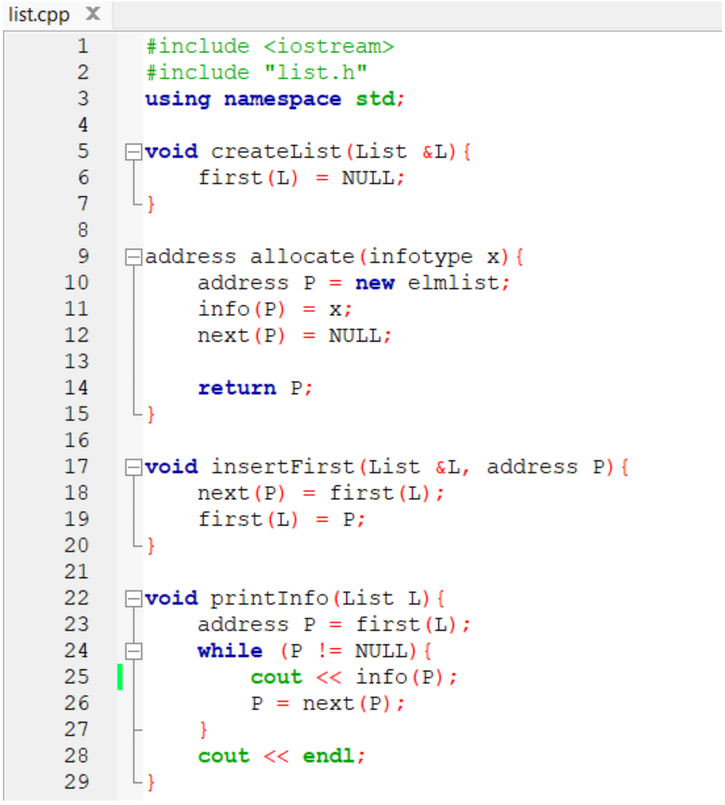
7. sesi have fun :

- list.h :

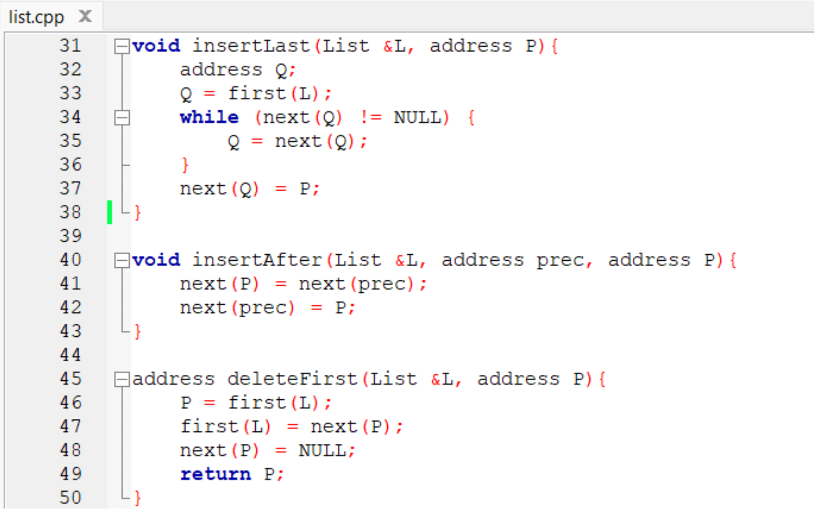


```
#include <iostream>
#define first(L) L.first
#define next(P) P->next
#define info(P) P->info
#define next(Q) Q->next
#define info(Q) Q->info
#define next(prec) prec->next
#define info(prec) prec->info
using namespace std;
typedef int infotype;
typedef struct elmlist *address;

struct elmlist{
    infotype info;
    address next;
};
struct List{
    address first;
};

void createList(List &L);
address allocate(infotype x);
void insertFirst(List &L, address P);
void printInfo(List L);
void insertLast(List &L, address P);
void insertAfter(List &L, address prec, address P);
address deleteFirst(List &L, address P);
address deleteLast(List &L, address P);
address deleteAfter(List &L, address prec, address P);
```

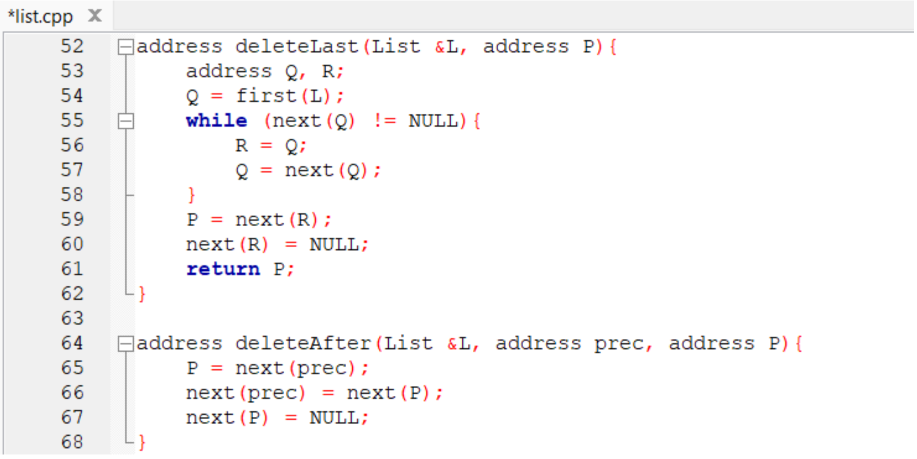
- list.cpp :



```
#include <iostream>
#include "list.h"
using namespace std;

void createList(List &L){
    first(L) = NULL;
}

address allocate(infotype x){
    address P = new elmlist;
    info(P) = x;
    next(P) = NULL;

    return P;
}

void insertFirst(List &L, address P){
    next(P) = first(L);
    first(L) = P;
}

void printInfo(List L){
    address P = first(L);
    while (P != NULL){
        cout << info(P);
        P = next(P);
    }
    cout << endl;
}
```

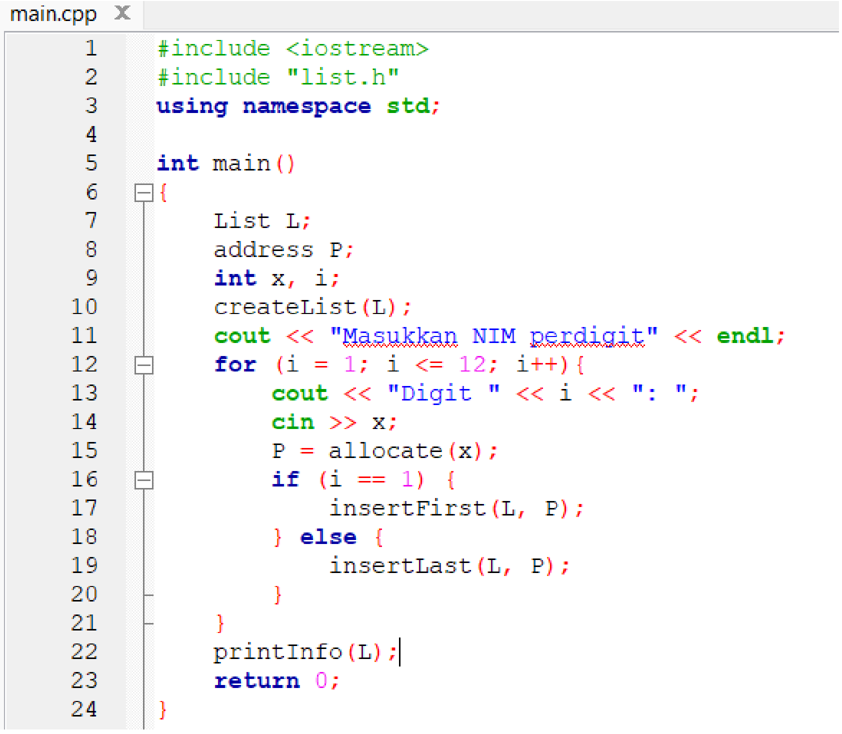


```cpp
void insertLast(List &L, address P){
    address Q;
    Q = first(L);
    while (next(Q) != NULL) {
        Q = next(Q);
    }
    next(Q) = P;
}

void insertAfter(List &L, address prec, address P){
    next(P) = next(prec);
    next(prec) = P;
}

address deleteFirst(List &L, address P){
    P = first(L);
    first(L) = next(P);
    next(P) = NULL;
    return P;
}
```



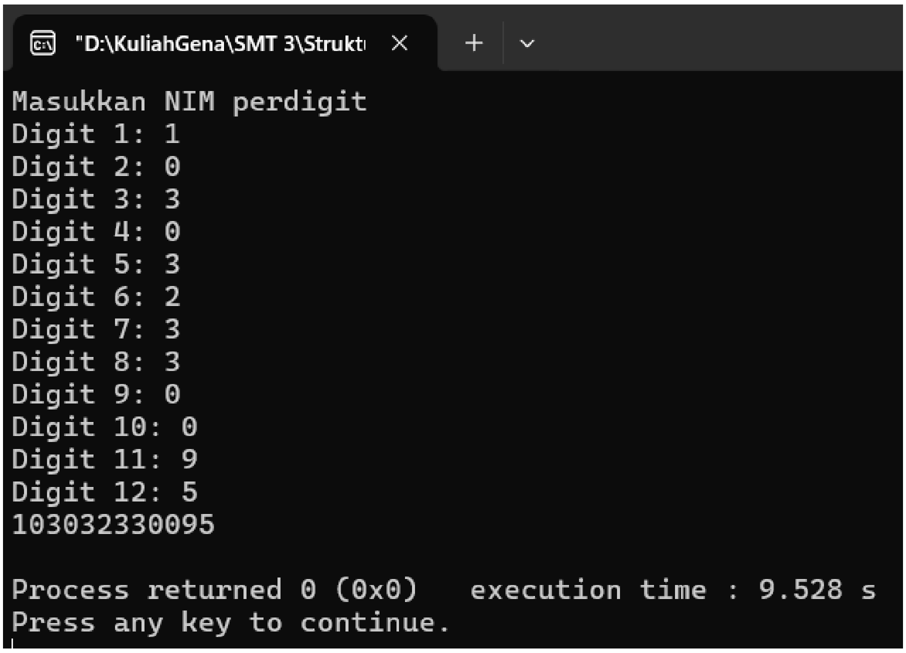
```cpp
address deleteLast(List &L, address P){
    address Q, R;
    Q = first(L);
    while (next(Q) != NULL){
        R = Q;
        Q = next(Q);
    }
    P = next(R);
    next(R) = NULL;
    return P;
}

address deleteAfter(List &L, address prec, address P){
    P = next(prec);
    next(prec) = next(P);
    next(P) = NULL;
}
```

- main.cpp :



```cpp
#include <iostream>
#include "list.h"
using namespace std;

int main()
{
    List L;
    address P;
    int x, i;
    createList(L);
    cout << "Masukkan NIM perdigit" << endl;
    for (i = 1; i <= 12; i++){
        cout << "Digit " << i << ": ";
        cin >> x;
        P = allocate(x);
        if (i == 1) {
            insertFirst(L, P);
        } else {
            insertLast(L, P);
        }
    }
    printInfo(L);
    return 0;
}
```

- output :



```
Masukkan NIM perdigit
Digit 1: 1
Digit 2: 0
Digit 3: 3
Digit 4: 0
Digit 5: 3
Digit 6: 2
Digit 7: 3
Digit 8: 3
Digit 9: 0
Digit 10: 0
Digit 11: 9
Digit 12: 5
103032330095

Process returned 0 (0x0)   execution time : 9.528 s
Press any key to continue.
```