Main.cpp:

```
#include "SLL.h"
  using namespace std;
  int main()
□ {
      List L;
      address p,q;
      CreateList_103032330054(L);
      int NUM = 0, pilihan = 0, x;
      infotype data;
      pilihan = selectMenu 103032330054();
      while (pilihan!=0) {
           switch (pilihan) {
               cin >> x;
               for (int i = 0; i < x; i++) {
                   cin >> data;
                   p = CreateNewElm 103032330054(data);
                   InsertLast 103032330054(L,p);
               break;
    case 2:
       Show 103032330054(L);
        break;
       p = findMin_103032330054(L);
        cin >> data;
        insertMiddle_103032330054(L, data);
    pilihan = selectMenu 103032330054();
```

SLL.cpp:

```
#include "SLL.h"
  void CreateList 103032330054(List &L) {
        L.first = NULL;
  ■address CreateNewElm 103032330054(infotype data){
       address p = new elmlist;
       p->next = NULL;
       p->info=data;
        return p;
  void InsertLast 103032330054(List &L, address p) {
       if (L.first==NULL) {
           L.first = p;
       }else{
            address q = L.first;
            while (q->next!=NULL) {
               q = q->next;
 •
           q->next = p;
_void Show_103032330054(List L){
    address p = L.first;
     while (p!=NULL) {
cout << p->info << ", ";
         p=p->next;
1
=address findMin 103032330054(List L) {
     address p = L.first;
     address q = L.first;
while (p!=NULL) {
         if (q->info > p->info) {
             q = p;
         p=p->next;
     return q;
```

```
■void insertMiddle 103032330054(List &L, int data) {
        address p,q;
        q=CreateNewElm 103032330054(data);
        int y = 1;
        p = L.first;
        while (p!=NULL) {
             x++;
            p=p->next;
 if (x%2==0) {
             x = x/2;
        }else{
             x = (x/2) +1;
        p = L.first;
        while (y!=x) {
             p=p->next;
             y++;
        q->next = p->next;
        p->next = q;
int selectMenu 103032330054(){
      cout << endl << "===== MENU =====" << endl;
cout << "1. Menambah N data baru" << endl;
cout << "2. Menampilkan semua data" << endl;</pre>
      int input = 0;
      cin >> input;
      return input;
```

SLL.h:

```
typedef int infotype;
typedef struct elmlist *address;
=struct elmlist{
      address next;
      infotype info;
L<sub>};</sub>
struct List{
     address first;
  void CreateList_103032330054(List &L);
  address CreateNewElm 103032330054(infotype data);
  void InsertLast_103032330054(List &L, address p);
  void Show 103032330054(List L);
  address findMin 103032330054(List L);
  void insertMiddle_103032330054(List &L, int data);
 int selectMenu 103032330054();
#endif // SLL H INCLUDED
```

Hasil:

```
==== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 1
Jumlah data yang akan ditambahkan: 5
Masukkan data baru: 40
Masukkan data baru: 90
Masukkan data baru: 30
Masukkan data baru: 70
Masukkan data baru: 50
==== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 2
40, 90, 30, 70, 50,
==== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 3
Data dengan nilai minimal: 30
```

```
Data dengan nilai minimal: 30
==== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 4
Masukkan data yang akan ditambahkan pada tengah list: 99
===== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 2
40, 90, 30, 99, 70, 50,
==== MENU =====
1. Menambah N data baru
2. Menampilkan semua data
3. Menampilkan data min
4. Menambahkan data tengah
0. Exit
Pilihan menu: 0
BYE BYE
Process returned 0 (0x0) execution time : 51.599 s
Press any key to continue.
```