```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct Mhs Mhs;  // memberi nama alias untuk struct Mhs
typedef struct MatKul MatKul;  // memberi nama alias untuk struct MatKul

struct MatKul {  // definisi struct MatKul
       char kode[7];
       char nama[30];
       int sks;
};

struct Mhs {  // definisi struct Mhs yang merupakan linked list
       char nim[15];
       char nama[50];
       int jumlah_matkul;
       MatKul matKul[7];  // deklarasi array variable matKul tiap mahasiswa
       Mhs *next;
};

Mhs *head;  // deklarasi pointer head

void init() {
     head = NULL;  // fungsi inisialisasi linked list
}

void printMhs() {
     // fungsi untuk menampilkan isi linked list Mhs
     Mhs *pWalker = head;
     int count = 1;
     while(pWalker != NULL){
         printf("Mahasiswa %d\n", count);
         printf("Nama\t: %s\n", pWalker->nama);
         printf("NIM\t: %s\n", pWalker->nim);
         printf("Matakuliah\t:\n");
         int i=0;
         for(i=0;i<pWalker->jumlah_matkul;i++){
             printf("\t%d\n", i);
             MatKul mk = pWalker->matKul[i];
             printf("\tNama\t: %s\n", mk.nama);
             printf("\tKode\t: %s\n", mk.kode);
             printf("\tsks\t: %d\n", mk.sks);

         }
         printf("-----------------------------------------------\n");
         pWalker = pWalker->next;
         count++;
     }
}

void freeMemory() {
     // fungsi untuk menghapus seluruh memory linked list
     Mhs *pWalker = head;
     Mhs *pDelete = NULL;
     while(pDelete != NULL){
         pDelete = pWalker;
```

```
        pWalker = pWalker->next;
        free(pDelete);
     }
}

int addMhs(char nama[50], char nim[15]) {
    // fungsi untuk menambah mahasiswa
    Mhs *pNew = (Mhs *) malloc(sizeof(Mhs));
    if(pNew == NULL){
            return 0;
    }

    strcpy(pNew->nama, nama);
    strcpy(pNew->nim, nim);


    if(head == NULL){
       head = pNew;
    }else{
       pNew->next = head;
       head = pNew;
    }

    return 1;
}

int addMatKul(char nimMhs[15], char namaMatKul[30], char kode[7], int sks) {
    // fungsi untuk menambah mata kuliah mahasiswa tertentu
    Mhs *pFind = head;
    while(pFind !=NULL && strcmp(pFind->nim,nimMhs)!=0){
          pFind = pFind->next;
    }

    if(pFind == NULL){
         return 0;
    }
    int idx = pFind->jumlah_matkul;
    MatKul mk = pFind->matKul[idx];
    strcpy(mk.nama, namaMatKul);
    strcpy(mk.kode, kode);
    mk.sks = sks;
    pFind->matKul[idx] = mk;
    pFind->jumlah_matkul = pFind->jumlah_matkul + 1;

    return 1;
}

void reverseMhsList() {
      Mhs *pCur = head;
      Mhs *pPre = NULL;
      Mhs *pTemp =  NULL;
      while(pCur != NULL){
            pTemp = pCur->next;
            pCur->next = pPre;
            pPre = pCur;
            pCur = pTemp;
      }
```

```
      head = pPre;
}

int main(int argc, char** argv) {
    init();
    addMhs("Ali", "091096001");
    addMhs("Budi", "091096002");
    addMatKul("091096002", "Sistem Operasi", "MTI002", 3);
    addMatKul("091096002", "Algoritma", "MTI001", 4);
    addMatKul("091096001", "Pemrograman Lanjut", "MTI003", 4);
    printMhs();
    printf("\n============== REVERSE ================\n");
    reverseMhsList();
    printMhs();
    freeMemory();
    system("PAUSE");
    return 0;
}
```