

```

// Chương trình thao tác trên Danh Sách Liên Kết
// Mỗi phần tử của DSLK là một con trỏ Data
// Ghi xuống file và đọc ngược từ file lean
// @ by Phạm Thế Bảo, khoa Toán – Tin học
// trường Đại Học Khoa Học Tự Nhiên Tp. HCM

```

```

#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<stdlib.h>
#define max 3 // max chính số phần tử tối đa của mảng
struct Data {
    char nameNV[80];
    unsigned int date[3];
    char address[100];
    char sex; // M: male, F: female, O: other
    unsigned int mark[max];
    char nameMark[max][30];
};
struct tagNode{
    Data *data;
    tagNode *Next,*Prev;
};
typedef tagNode *Node;
struct LinkedList{
    Node Head,Tail;
};
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void printData(Data *d){
    printf("Ho ten : %s\n",d->nameNV);
    printf("Ngày tháng nam sinh: %d/%d/%d",d->date[0],d->date[1],d->date[2]);
    printf("\nDia chi: %s",d->address);
    printf("\nGioi tinh: %c",d->sex);
    printf("\nDiem cac mon hoc:");
    for(int i=0;i<max;i++)
        printf("\n\t Mon %s co diem %d",d->nameMark[i],d->mark[i]);
    }
    //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void inputData(Data &d){
    fflush();
    printf("\nNhap ho ten: ");gets(d.nameNV);
    printf("Nhap ngay sinh: ");scanf("%d",&d.date[0]);
    printf("Nhap thang sinh: ");scanf("%d",&d.date[1]);
    printf("Nhap nam sinh: ");scanf("%d",&d.date[2]);
    fflush();
    printf("Nhap dia chi: ");gets(d.address);
    printf("Nhap gioi tinh [(M/F/O) - F:female - M:male - O:other] : ");scanf("%c",&d.sex);
}

```

```

for(int i=0;i<max;i++){
    flushall();
    printf("Nhap ten mon hoc thu %d: ",i+1);gets(d.nameMark[i]);
    printf("Nhap diem mon hoc thu %d ",i+1);scanf("%d",&d.mark[i]);
}
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

Node makeNode(Data d){
    Data *t = new Data;
    if(!t)return NULL;
    strcpy(t->nameNV,d.nameNV);
    for(int i=0;i<3;i++)t->date[i] = d.date[i];
    strcpy(t->address,d.address);
    t->sex=d.sex;
    for(i=0;i<max;i++){
        t->mark[i] = d.mark[i];
        strcpy(t->nameMark[i],d.nameMark[i]);
    }
    Node p = new tagNode;
    if(!p)return NULL;
    p->data = t;
    p->Next = p->Prev = NULL;
    return p;
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

void initList(LinkedList &l){
    l.Head = l.Tail = NULL;
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

int insertHead(LinkedList &l, Data d){
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

```

int insertTail(LinkedList &l, Data d){
    Node p = makeNode(d);
    if(!p){
        printf("Not enough memory !!!");
        return -1;
    }
    if(!l.Head)l.Head = l.Tail = p;
    else{
        l.Tail->Next = p;
        p->Prev = l.Tail;
        l.Tail = p;
    }
}

```

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int insertTail(LinkedList &l, Data *d){
    Node p = new tagNode;
    if(!p)return NULL;
    p->data = d;
    p->Next = p->Prev = NULL;
    if(!p){
        printf("Not enough memory !!!");
        return -1;
    }
    if(!l.Head)l.Head = l.Tail = p;
    else{
        l.Tail->Next = p;
        p->Prev = l.Tail;
        l.Tail = p;
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int insertAfter(LinkedList &l, Node p,Data d){
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int deleteHead(LinkedList &l){
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int deleteTail(LinkedList &l){
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
int deleteAfter(LinkedList &l,Node p){
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void deleteList(LinkedList &l){
    Node p = l.Head;
    while(p){
        l.Head = p->Next;
        p->Next = NULL;
        delete p;
        p = l.Head;
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void printList(LinkedList l){
    printf("\n");

    Node p = l.Head;
    while(p){
        printData(p->data);
}

```

```

        p=p->Next;
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void readFromFile(LinkedList &l, char *st){
    FILE *f;
    if ((f = fopen(st, "rb"))== NULL){
        fprintf(stderr, "Cannot open input file.\n");
        exit(1);
    }
    Data *d;
    while (!feof(f)){
        fread(d,sizeof(Data),1,f);
        if(!feof(f))insertTail(l,d);
    }
    fclose(f);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void write2File(LinkedList l,char *st){
    FILE *f;
    if ((f = fopen(st, "wb"))== NULL){
        fprintf(stderr, "Cannot open output file.\n");
        exit(1);
    }
    Node p = l.Head;
    while(p){
        fwrite(p->data,sizeof(Data),1,f);
        p=p->Next;
    }
    fclose(f);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
void main(){
    clrscr();
    LinkedList list;
    initList(list);
    Data d;
    readFromFile(list,"c:\\temp\\list.dat");
    //inputData(d);
    //insertTail(list,d);
    printList(list);
    //write2File(list,"c:\\temp\\list.dat");
    getch();
    deleteList(list);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```