

BÀI TẬP DS NỐI ĐƠN

Bài toán 1: Chương trình quản lý sinh viên (mã SV, họ tên, năm sinh, điểm tổng kết) bằng danh sách nối đơn với các chức năng

- Khai báo cấu trúc dữ liệu
- Tạo mới danh sách
- Hiển thị danh sách
- Xác định chiều dài danh sách
- Tìm kiếm sinh viên theo mã và hiển thị thông tin của sinh viên nếu tìm thấy.

Hướng dẫn:

```
//kb ctdl của ds
struct SV{
    char maSV[15];
    char hoTen[30];
    int namSinh;
    float diemTK;
};
struct Node{
    SV infor;
    Node *next;
};
typedef Node *TRO;
```

-

Bài toán 2: Cho Danh sách móc nối đơn lưu trữ thông tin về sinh viên gồm: mã sinh viên, họ và tên, giới tính, năm sinh, điểm tổng kết.

Viết chương trình thực hiện các yêu cầu sau:

- Khai báo cấu trúc dữ liệu của danh sách.
- Tạo một Danh sách sinh viên có từ 5 phần tử trở lên.
- Hiển thị danh sách sau khi tạo

- Hiển thị danh sách sinh viên có giới tính “Nữ” trong danh sách.
- Tìm sinh viên có điểm trung bình lớn nhất trong danh sách.
- Xóa những sinh viên có điểm TB bằng điểm TB lớn nhất trong DS. Hiển thị DS sau khi xóa.

Hướng dẫn:

1. Khai báo cấu trúc dữ liệu của danh sách:

```
//kb ctdl của ds
struct SV{
    char maSV[15];
    char hoTen[30];
    char gioiTinh[5];
    int namSinh;
    float diemTB;
};
struct Node{
    SV infor;
    Node *next;
};
typedef Node *TRO;
//khởi tạo ds rỗng
void creat(TRO &L){
    L=NULL;
}
```

2. Khởi tạo ds rỗng, kiểm tra ds đầy, kiểm tra ds rỗng, thêm sv vào cuối danh sách.

```
//khởi tạo ds rỗng
void creat(TRO &L){
    L=NULL;
}
//kiểm tra ds rỗng
int Empty(TRO L){
    return L==NULL;
}
```

3. Nhập và xuất thông tin DS sinh viên

//nhập/xuất thông tin 1 sv

```

//nhap ttin cho 1 sv
void NhapSV(SV &sv){
    cout<<"\tNhap ma SV: ";fflush(stdin);gets(sv.maSV);
    cout<<"\tNhap ho ten SV: ";fflush(stdin);gets(sv.hoTen);
    cout<<"\tNhap gioi tinh: ";fflush(stdin);gets(sv.gioiTinh);
    cout<<"\tNhap nam sinh: ";cin>>sv.namSinh;
    cout<<"\tNhap diem TK: "; cin>>sv.diemTB;
}

```

```

//xuat ttin cho 1 sv

```

```

void XuatSV(SV sv){
    cout<<setw(10)<<sv.maSV;
    cout<<setw(20)<<sv.hoTen;
    cout<<setw(20)<<sv.gioiTinh;
    cout<<setw(10)<<sv.namSinh;
    cout<<setw(10)<<sv.diemTB<<endl;
}

```

```

//nhap ttin cho 1 sv

```

```

void NhapSV(SV &sv){
    cout<<"\tNhap ma SV: ";fflush(stdin);gets(sv.maSV);
    cout<<"\tNhap ho ten SV: ";fflush(stdin);gets(sv.hoTen);
    cout<<"\tNhap gioi tinh: ";fflush(stdin);gets(sv.gioiTinh);
    cout<<"\tNhap nam sinh: ";cin>>sv.namSinh;
    cout<<"\tNhap diem TK: "; cin>>sv.diemTB;
}

```

```

//Nhap danh sách SinhVien (>=5)

```

```

//tao nut moi

```

```

Node* makenode(SV s)
{
    TRO p = new Node;
    if (p == NULL)
    {
        cout << "\nKhong du bo nho.";
        return NULL;
    }
    strcpy(p->infor.maSV, s.maSV);
    strcpy(p->infor.hoTen, s.hoTen);
    p->infor.diemTK = s.diemTK;
    p->next = NULL;
    return p;
}

```

```

void NhapDSSV(TRO &L){
    creat(L);
    SV x;
    int n;
    TRO P, Q;
    int i=1;
    do{
        cout<<"Nhap so luong sv: "<<endl;
        cin>>n;
        if(n<=0) cout<<"So luong sv phai >0, yc nhap lai:";
    }while(n<=0);
    for(i = 0; i<n;i++){
        TRO p;
        cout<<"Nhap thong tin sv thu "<<i+1<<endl;
        NhapSV(x);
        p = makenode(x);
        AddLast(L,x);
    }
}

```

//Hien thi DSSV

//Hien thi DSSV

```

void HienthiDSSV(TRO L){
    if(L==NULL){
        cout<<"DS rong "<<endl;
        return;
    }
    cout<<setw(5)<<"STT"<<setw(10)<<"Ma SV"<<setw(20)<<"Ho ten"<<setw(10)<<"NSinh"
    <<setw(10)<<"Diem TK"<<endl;
    TRO Q = L;
    int i=0;
    while(Q!=NULL){
        SV x = Q->infor;
        cout<<setw(5)<<(1+i);
        cout<<setw(10)<<x.maSV;
        cout<<setw(20)<<x.hoTen;
        cout<<setw(10)<<x.namSinh;
        cout<<setw(10)<<x.diemTK<<endl;
        Q = Q->next;
    }
}

```

4. Hiển thị những sinh viên có giới tính “Nữ”.

```
//hien thi sv co gioi tinh: Nu
void hienthiSV_Nu(TRO L){
    if(L==NULL){
        cout<<"DS rong"<<endl;
        return;
    }
    cout<<setw(5)<<"STT"<<setw(10)<<"Ma SV"<<setw(20)<<"Ho ten"
    <<setw(10)<<"NSinh"<<setw(10)<<"Diem TK"<<endl;
    TRO Q = L;
    int i=0;
    while(Q!=NULL){
        SV x = Q->infor;
        if(strcmp(x.gioiTinh,"Nu")==0){
            cout<<setw(5)<<(1+i);
            cout<<setw(10)<<x.maSV;
            cout<<setw(20)<<x.hoTen;
            cout<<setw(20)<<x.gioiTinh;
            cout<<setw(10)<<x.namSinh;
            cout<<setw(10)<<x.diemTB<<endl;
            i++;
        }
        Q=Q->next;
    }
    if(i==0) cout<<"Ko co sv nao co ma 1004 trong DS"<<endl;
}
```

5. Tìm SV có ĐTB max, và xóa những SV có dtb = dtb max

```
//SV co dtb max
float DTB_max(TRO L){
    if(L==NULL){
        cout<<"DS rong"<<endl;
        //return;
    }
    TRO Q = L;
    int i = 0;
    float m = Q->infor.diemTB;
    while(Q!=NULL){
        SV x = Q->infor;
        if(Q->infor.diemTB>m)
            m = Q->infor.diemTB;
        Q = Q->next;
    }
    return m;
}
```

```

//xoa SV có dtb = dtb max
void xoaDTB(TRO &L){
    float m = DTB_max(L);
    TRO Q = L;
    while(L!=NULL&&L->infor.diemTB==m){
        L = L->next;
        delete Q;
    }
    TRO P = L;
    while(P!=NULL&&P->next!=NULL){
        if(P->next->infor.diemTB==m){
            TRO Q = P->next;
            P->next = Q->next;
            delete Q;
        }
        else P = P->next;
    }
}

```

6. Hàm xoá phần tử đầu tiên trong DS

```

//xoa sv dau tien trong ds
void firstDelete(TRO &L){
    TRO Q;
    Q = L;
    L = L->next;
    delete Q;
}

```

Bài toán 3: Cho Danh sách móc nối đơn lưu trữ thông tin về hàng hóa gồm: mã hàng, tên hàng, đơn giá, số lượng, thành tiền. (thành tiền = số lượng * đơn giá).

Viết chương trình thực hiện các yêu cầu sau:

- Khai báo cấu trúc dữ liệu của danh sách.
- Tạo một Danh sách hàng hóa gồm 5 phần tử dữ liệu
- Hiển thị danh sách hàng hóa sau khi tạo
- Hiển thị các hàng hóa có số lượng lớn nhất trong danh sách.
- Chèn thêm một hàng hóa mới vào vị trí thứ 3 trong danh sách. Hiển thị danh sách sau khi chèn.

Hướng dẫn

1. Tạo một ds gồm 5 pt dữ liệu

```
//Nhap DSHH - gom 5pt
void input_List(TRO &L){
    TRO Q, P;
    HH x;
    create(L);
    int i = 1;
    do{
        cout<<"Nhap HH thu "<<i<<": "<<endl;
        nhapHH(x);
        P = new Node;
        P->infor = x;
        P->next = NULL;
        if(L==NULL){
            L = P;
        }
        else Q->next = P;
        Q = P;
        i++;
    }while(i<6);
}
```

2. Hiện các hàng hóa có số lượng lớn nhất

```
//tim soLuong MAX
int soLuong_max(TRO L){
    if(L==NULL){
        cout<<"DS rong"<<endl;
        //return;
    }
    TRO Q = L;
    int i = 0;
    int m = Q->infor.soLuong;
    while(Q!=NULL){
        HH x = Q->infor;
        if(Q->infor.soLuong>m)
            m = Q->infor.soLuong;
        Q = Q->next;
    }
    return m;
}
```



```

//hien DSHH co SL max
void output_SL_max(TRO L){
    if(L==NULL){
        cout<<"DS rong"<<endl;
        return;
    }
    int m = soLuong_max(L);
    TRO Q = L;
    int i = 0;
    //xuat tieu de
    cout<<"STT"<<setw(20)<<"Ma hang"<<setw(20)<<"Ten hang"<<setw(20)<<"Don gia"
    <<setw(20)<<"So luong"<<setw(20)<<"Thanh tien"<<endl;
    while(Q!=NULL){
        HH x = Q->infor;
        if(x.soLuong==m){
            cout<<i+1<<setw(20)<<x.maH<<setw(20)<<x.tenH<<setw(20)<<x.donGia
            <<setw(20)<<x.soLuong<<setw(20)<<x.soLuong*x.donGia<<endl;
            i++;
        }
        Q = Q->next;
    }
}

```

3. Chèn hàng hóa mới vào vị trí thứ 3

```

//bs vao dau DS
void firstAdd(TRO &L, HH x){
    TRO P;
    P = new Node;
    P->infor = x;
    P->next = L;
    L = P;
}

//bs vao sau 1 nut
void insert(TRO &L, TRO Q, HH x){
    TRO P;
    P = new Node;
    P->infor = x;
    P->next = Q->next;
    Q->next = P;
}

```



```

//chen HH moi vao vt thu 3
void boSung(TRO &L){
    int k=3;
    HH x;
    cout<<"Nhap thong tin hang hoa can chen: "<<endl;
    nhapHH(x);
    if(k==1) firstAdd(L,x);
    else{
        TRO Q = L;
        int d = 1;
        while(d<k-1){
            d++;
            Q = Q->next;
        }
        insert(L,Q,x);
    }
    cout<<"\n DS HH sau khi chen vao vt 3====\n";
    output_List(L);
}

```