

BÀI THỰC HÀNH SỐ 1

DANH SÁCH TỔ CHỨC BẰNG MẢNG

1. Mục đích và yêu cầu của bài thực hành

- Bài thực hành số 1 mục đích cụ thể hóa những kiến thức về tổ chức dữ liệu kiểu mảng để lưu trữ và thao tác với danh sách. Qua bài thực hành sinh viên được thao tác với một danh sách cụ thể với những thao tác cụ thể và gần với thực tế. Để hoàn thành bài thực hành sinh viên sẽ thấy cần bổ sung thêm những thao tác cần thiết nào mà trong khi học lý thuyết không trình bày.
- Bài thực hành cũng rèn luyện cho sinh viên kỹ năng lập trình, nhất là những kỹ năng làm việc với mảng.
- Bài thực hành yêu cầu sinh viên cài đặt được một số thao tác cơ bản trên danh sách tổ chức bằng mảng và hoàn chỉnh các thao tác cần thiết khác để hoàn chỉnh thành một chương trình chạy được.

2. Các kiến thức cần thiết

Để hoàn thành bài thực hành, sinh viên cần có các kiến thức và kỹ năng sau:

- Sử dụng được một công cụ lập trình.
- Sử dụng được một ngôn ngữ lập trình hướng cấu trúc. Biết cách tổ chức chương trình theo các đơn thể và sử dụng được các đơn thể.
- Kiến thức về tổ chức danh sách bằng mảng và các thao tác cơ bản: khởi tạo, thêm, xóa, tìm kiếm, sắp xếp.

3. Nội dung bài thực hành

Cần quản lý một danh bạ điện thoại của các khách hàng trong đó mỗi khách hàng gồm các thông tin: tên khách hàng, địa chỉ, các số điện thoại của khách hàng (mỗi khách hàng không quá 10 số điện thoại, giả sử tên khách hàng không trùng nhau).

Cho trước cách tổ chức dữ liệu kiểu mảng để lưu danh bạ với các thao tác cơ bản đã cài đặt bằng ngôn ngữ lập trình C:

```
#define max 100
// Kiểu ThueBao biểu diễn các đối tượng khách hàng là các thuê bao.
struct ThueBao
{
    char tenTB[30];
    char diaChi[50];
    char soDT[10][11];
    int soSoDT;
};
struct DanhBa
{
    struct ThueBao db[max];
    int soTB;
};
```

3.1. Phân thực hành tại phòng máy:

Viết các hàm thực hiện:

1. Nhập một thuê bao từ bàn phím.
2. In một thuê bao lên màn hình.
3. Nhập một danh sách các thuê bao.
4. In danh sách các thuê bao lên màn hình.
5. Thêm một thuê bao vào vị trí k của danh sách.
6. Xóa một thuê bao tại vị trí k trong danh sách.
7. Tìm một thuê bao khi biết tên khách hàng. Kết quả trả về vị trí của thuê bao tìm được, nếu không tìm được thì trả về -1.

8. Xóa một thuê bao theo tên khách hàng.
9. Tìm một thuê bao theo số điện thoại. Kết quả trả về vị trí của thuê bao tìm được, nếu không tìm được thì trả về -1.
10. Sắp xếp danh bạ theo thứ tự tăng của tên thuê bao.
11. Viết chương trình thực hiện các chức năng trên.

3.2 Hướng dẫn:

- Hàm nhập 1 thuê bao như sau:

```
void nhapThueBao(struct ThueBao *x)
{
    int i;
    printf("Nhap ten thue bao:"); gets(x->tenTB);
    printf("Nhap dia chi thue bao:"); gets(x->diaChi);
    printf("Nhap so so dien thoai cua thue bao:");
    scanf("%d",&x->soSoDT);
    fflush(stdin);
    for (i=0; i<x->soSoDT; i++)
    {
        printf("Nhap so dien thoai thu %d ", i, " : "); gets(x->soDT[i]);
    }
}
```

Tương tự cho hàm in một thuê bao.

- Hàm nhập một danh sách thuê bao:

```
void nhapDanhBa(struct DanhBa *d)
{
    int i;
    printf("So thue bao: "); scanf("%d",&d->soTB);
    fflush(stdin);
    for(i=0; i<d->soTB;i++)
        nhapThueBao(&d->db[i]);
}
```

Tương tự cho hàm in danh sách thuê bao.

Hàm thêm 1 thuê bao vào danh bạ tại vị trí k:

```
void themThueBao(struct DanhBa *d, struct ThueBao x, int k)
{
    int i,j;
    for(i=d->soTB; i>k; i--)
        d->db[i]=d->db[i-1];
    d->db[k] = x;
    d->soTB++;
}
```

- Chương trình chính:

```
void main()
{
    struct DanhBa db;
    struct ThueBao t;
    nhapDanhBa (&db);
    inDanhBa (db);
    nhapThueBao (&t);
    themThueBao (&db,t,1);
    inDanhBa (db);
}
```

7. Nhắc lại: Để so sánh hai chuỗi ký tự bằng nhau trong c dùng hàm: strcmp(xau1,xau2)==0
8. Sử dụng hàm tìm thuê bao theo tên thuê bao và hàm xóa thuê bao tại vị trí k ở trên.
 - + Tìm thuê bao theo họ tên kết quả của hàm là k
 - + Nếu k!=-1 thì xóa thuê bao tại vị trí k

```

-----
int tim(struct DanhBa d, char ten[])
{
    ...
}
void xoa(struct DanhBa *d, int k)
{
    ...
}
void tim_xoa(struct DanhBa *d, char ten[])
{
    int k;
    k = tim(d,ten);
    if(k!=-1) xoa(d,k);
}

```

Trong hàm main()

- Nhập danh bạ db
- Nhập tên cần xóa t
- Gọi hàm tim_xoa(&db,t);
- In lại danh bạ db

9.

Thuật toán:

- Duyệt lần lượt từng thuê bao trong danh bạ:
 - + Với thuê bao đang xét duyệt lần lượt số điện thoại của thuê bao:

Nếu số điện thoại đang xét trùng với số điện thoại cần tìm thì trả về vị trí của

thuê bao

- Trả về -1

```

int timSDT(struct DanhBa d, char sdt[])
{
    for(i=0; i<d.soTB; i++)
        for(j=0; j<d.tb[i].soSDT; j++)
            if(strcmp(d.tb[i].soSDT[j],sdt)==0)
                return i;
    return -1;
}

```

3.3. Phần thực hành ở nhà:

1. Thêm một số điện thoại vào danh sách các số điện thoại của khách hàng có tên x (nếu có).
2. Xóa một số điện thoại x của khách hàng nào đó.
3. Liệt kê tên các khách hàng sử dụng mạng điện thoại Viettel, biết rằng số điện thoại của Viettel bắt đầu bằng 098 hoặc 097.
4. Thêm một thuê bao x vào danh bạ, nếu chưa có thì thêm vào vị trí thích hợp của danh bạ sao cho danh sách vẫn theo thứ tự tăng của tên khách hàng, nếu đã có thì bổ sung các số điện thoại chưa có vào thuê bao đã có trong danh bạ.
5. Viết chương trình sử dụng các chức năng trên.
