

CẤU TRÚC DỮ LIỆU

❖ KHÁI NIỆM

- Là cách tổ chức lưu trữ dữ liệu trong máy tính để mô hình hóa các điều kiện của bài toán, từ đó có thể sử dụng dữ liệu một cách hiệu quả.
- Cấu trúc dữ liệu có vai trò rất quan trọng trong việc giải quyết vấn đề bằng máy tính.

CẦU TRÚC DỮ LIỆU

❖ Ví dụ 1:

Viết chương trình nhập vào một danh sách gồm thông tin của cá nhân với họ và tên, năm sinh và giới tính. In ra danh sách cá nhân theo dạng như sau:

Truong Van Minh, 1998, Nam

Hoang Thi Thu Trang, 1998, Nu

```
#include <iostream>
using namespace std;
#define MAX_LIST 30
#define MAX_INFO 100
typedef char Canhan[MAX_INFO];
void nhap(Canhan ds[], int *sl) {
    int i;
    cout << "So luong nguoi: ";
    cin >> *sl; cin.ignore();
    for (i = 0; i < *sl; i++) {
        cout << "Nguoi thu " << i << ": ";
        cin.getline(ds[i], MAX_INFO);
    }
}
```

```
void xuat(Canhan ds[], int sl) {  
    int i;  
    cout << "Co " << sl << " nguoi" << endl;  
    for (i = 0; i < sl; i++)  
        cout << ds[i] << endl;  
}  
  
int main() {  
    Canhan nhom[MAX_LIST];  
    int n;  
    nhap(nhom, &n);  
    xuat(nhom, n);  
    return 0;  
}
```

CẦU TRÚC DỮ LIỆU

❖ Ví dụ 2:

Với yêu cầu như Ví dụ 1, giả sử có thêm yêu cầu:

Cho phép nhập vào giá trị t là số tuổi và in ra tất cả thông tin họ và tên, tuổi của các cá nhân có tuổi lớn hơn.

```
//...cac include nhu Vi du 1
#include <cstdlib>
#include <cstring>

//...cac khai bao, ham nhu Vi du 1
char* layHoTen(Canhan c) {
    char *p = strtok(c,",");
    return p;
}

int layTuoi(Canhan c) {
    int t;
    char *p = strtok(c, ",");
    p = strtok(NULL, ",");
    t = 2018 - atoi(p);
    return t;
}
```

```
void xuatTuoi(Canhan ds[], int sl, int t) {  
    char *hoten;  
    int tuoi, i;  
    for (i = 0; i < sl; i++) {  
        tuoi = layTuoi(ds[i]);  
        if (tuoi >= t) {  
            hoten = layHoTen(ds[i]);  
            cout << hoten << ", " << tuoi << endl;  
        }  
    }  
}
```

// ham main thay the cho ham main trong Vi du 1

```
int main() {  
    Canhan nhom[MAX_LIST];  
    int n, t;  
    nhap(nhom, &n);  
    xuat(nhom, n);  
    cout << "Nhap tuoi ";  
    cin >> t;  
    xuatTuoi(nhom, n, t);  
    return 0;  
}
```


CẤU TRÚC DỮ LIỆU

❖ Bài tập:

Xây dựng cấu trúc dữ liệu phù hợp để giải quyết các yêu cầu trong Ví dụ 2

CẦU TRÚC DỮ LIỆU

❖ TIÊU CHUẨN ĐÁNH GIÁ

- Phản ánh đúng thực tế
- Phù hợp với thao tác tìm kiếm, truy xuất, cập nhật, thêm hoặc xóa.
- Tiết kiệm tài nguyên hệ thống.
- Đơn giản và dễ hiểu.