



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

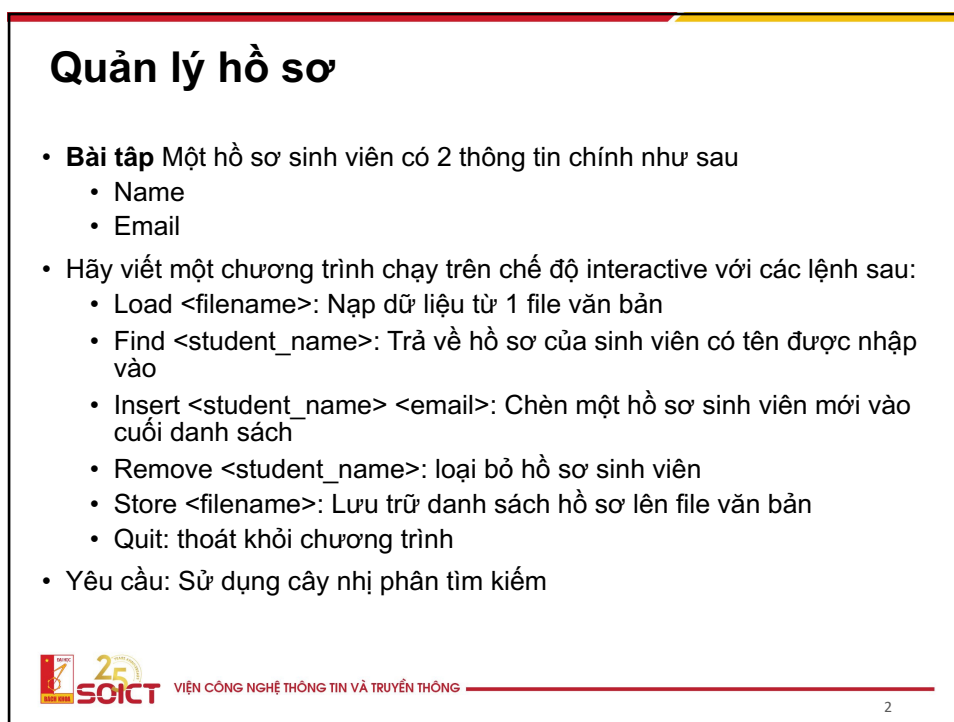
# LẬP TRÌNH C CƠ BẢN

## Tìm kiếm – phần 2

1

### Quản lý hồ sơ

- **Bài tập** Một hồ sơ sinh viên có 2 thông tin chính như sau
  - Name
  - Email
- Hãy viết một chương trình chạy trên chế độ interactive với các lệnh sau:
  - Load <filename>: Nạp dữ liệu từ 1 file văn bản
  - Find <student\_name>: Trả về hồ sơ của sinh viên có tên được nhập vào
  - Insert <student\_name> <email>: Chèn một hồ sơ sinh viên mới vào cuối danh sách
  - Remove <student\_name>: loại bỏ hồ sơ sinh viên
  - Store <filename>: Lưu trữ danh sách hồ sơ lên file văn bản
  - Quit: thoát khỏi chương trình
- Yêu cầu: Sử dụng cây nhị phân tìm kiếm



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

2

## Quản lý hồ sơ

```
#include <stdio.h>
#define MAX_L 256
#define MAX 100000

typedef struct Node{
    char name[256];
    char email[256];
    struct Node* leftChild;
    struct Node* rightChild;
}Node;
Node* root;
```



3

## Quản lý hồ sơ

```
void freeTree(Node* r){
    if(r == NULL) return;
    freeTree(r->leftChild);
    freeTree(r->rightChild);
    free(r);
}

void load(char* filename){
    FILE* f = fopen(filename,"r");
    if(f == NULL) printf("Load data -> file not found\n");
    root = NULL;
    while(!feof(f)){
        char name[256], email[256];
        fscanf(f,"%s%s",name, email);
        root = insert(root,name,email);
    }
    fclose(f);
}
```



7

## Quản lý hồ sơ

```
void inOrder(Node* r){
    if(r == NULL) return;
    inOrder(r->leftChild);
    printf("%s, %s\n",r->name,r->email);
    inOrder(r->rightChild);
}

void inOrderF(Node* r, FILE* f){
    if(r == NULL) return;
    inOrderF(r->leftChild,f);
    fprintf(f,"%s  %s\n",r->name,r->email);
    inOrderF(r->rightChild,f);
}

void printList(){
    inOrder(root);
    printf("\n");
}
```



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

8

8

## Quản lý hồ sơ

```
void processStore(){
    char filename[256];
    scanf("%s",filename);
    FILE* f = fopen(filename,"w");
    inOrderF(root,f);
    fclose(f);
}

void processInsert(){
    char name[256], email[256];
    scanf("%s%s",name,email);
    root = insert(root,name,email);
}

void processRemove(){
    char name[256];
    scanf("%s",name);
    root = removeStudent(root,name);
}
```



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

9

9

## Quản lý hồ sơ

```
void main(){
    while(1){
        printf("Enter command: ");
        char cmd[256];
        scanf("%s",cmd);
        if(strcmp(cmd,"Quit")==0) break;
        else if(strcmp(cmd,"Load")==0) processLoad();
        else if(strcmp(cmd,"Print")==0) printList();
        else if(strcmp(cmd,"Find")==0) processFind();
        else if(strcmp(cmd,"Insert")==0) processInsert();
        else if(strcmp(cmd,"Remove")==0) processRemove();
        else if(strcmp(cmd,"Store")==0) processStore();
    }
    freeTree(root);
}
```



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

10

10



11