HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

C PROGRAMMING LANGUAGE

ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

C PROGRAMMING LANGUAGE

BÀI 13: CẤU TRÚC (STRUCT)

ONE LOVE. ONE FUTURE.

NỘI DUNG

- Khái niệm cấu trúc
- Khai báo cấu trúc, biến cấu trúc
- Truy nhập cấu trúc
- Định nghĩa kiểu
- So sánh cấu trúc
- Mảng cấu trúc
- Ví dụ
- Cấu trúc và hàm
- Ví dụ
- Bài tập

KHÁI NIỆM CẤU TRÚC

- Thực tế chúng ta thường thao tác với các thực thể/đối tượng bao gồm một số thông tin đi theo nhóm. Ví dụ, số phức bao gồm giá trị phần thực và phần ảo, bản ghi một sinh viên chứa thông tin về họ tên, mã số, ngày sinh, điểm học phần, điểm gpa, điểm cpa, ...
- Mỗi thực thể như thế này được cấu tạo từ nhiều biến, nhưng có thể quản lí như một đơn vị logic.
- Một cấu trúc (struct) là một tập hợp của một số biến cùng hoặc khác kiểu dữ liệu, được đóng gói vào một khối hợp nhất.
- Một khai báo cấu trúc cho phép định một kiểu dữ liệu phức hợp cho các biến kiểu cấu trúc.
- Tên của các biến trong một cấu trúc còn được gọi là trường cấu trúc.

KHAI BÁO CẤU TRÚC, BIẾN CẤU TRÚC

Khai báo (kiểu) cấu trúc:

```
struct complex{
        int real;
        int img;
};

struct studentRec{
        char name[30];
        int mark;
};
```

Khai báo tên của cấu trúc

Các trường của cấu trúc

Đừng quên dấu ; sau khai báo cấu trúc

Khai báo một biến cấu trúc:

```
struct complex num;
struct studentRec john;
```

Tên kiểu cấu trúc

Tên của biến

Kết hợp khai báo kiểu và biến cấu trúc:

```
struct complex{
        int real;
        int img;
}num;
struct studentRec{
        char name[80];
        int mark;
}john;
```

TRUY NHẬP CẤU TRÚC

- Để truy nhập vào một trường của biến cấu trúc ta dùng toán tử '.'

Khởi tạo các trường cấu trúc giống như đối với các biến thông thường

```
struct studentRec john;

strcpy(john.name, "John");
john.mark = 7;

printf("%s co diem la %d", john.name, john.mark);
```

- Để truy nhập vào một trường của biến cấu trúc thông qua con trỏ cấu trúc

Khai báo một con trỏ có kiểu cấu trúc

```
struct studentRec john;
struct studentRec *ptr = &john;

strcpy(ptr->name, "John");
ptr->mark = 7;
printf("%s co diem la %d", ptr->name, ptr->mark);
```

ĐỊNH NGHĨA KIỂU (typedef)

- Dùng typedef để định nghĩa tên một kiểu dữ liệu mới mà có thể dùng trong các khai báo biến

```
struct studentRec{
   char name[80];
   int mark;
};

typedef struct studentRec Student;

Student studA, studB, *ptr;
Student stud_list[100];
```

Cấu trúc dữ liệu hiện có

Tên kiểu mới

Khai báo biến, con trỏ hay mảng với kiểu dữ liệu mới

Gộp định nghĩa kiểu với khai báo cấu trúc

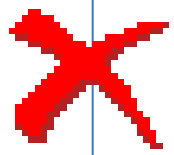
```
typedef struct studentRec{
   char name[80];
   int mark;
}Student;

Student studA, studB, *ptr;
Student stud_list[100];
```

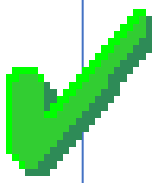
SO SÁNH CẤU TRÚC

- Không thể so sánh hai cấu trúc bằng toán tử ==
- Chỉ có thể so sánh từng trường của cấu trúc

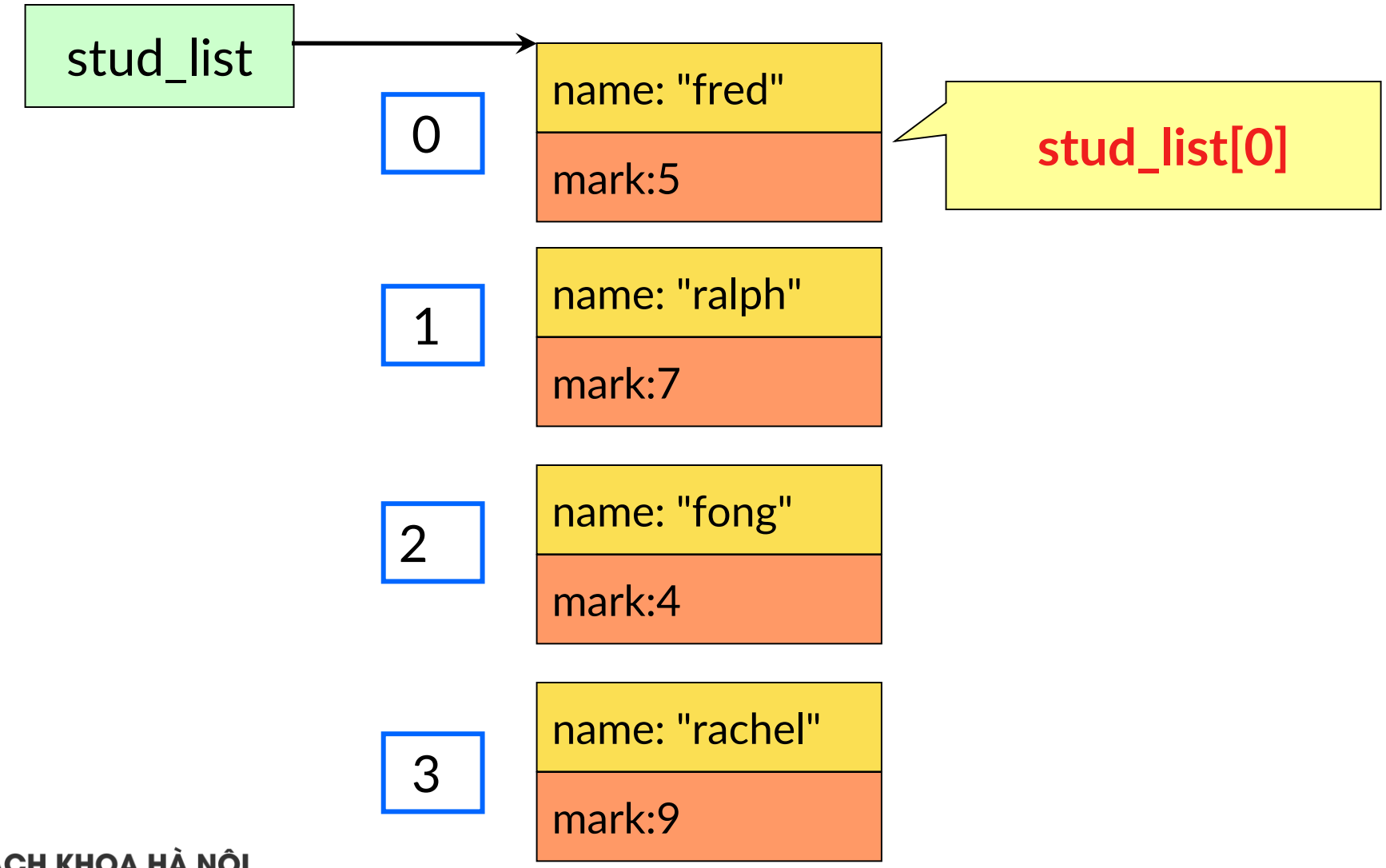
```
if(studA == studB){
    printf("Du lieu trung nhau.\n");
}
```



```
if((strcmp(studA.name,studB.name)== 0)
  &&(studA.mark == studB.mark)){
    printf("Du lieu trung nhau.\n");
}
```



MẢNG CẤU TRÚC



VÍ DỤ

```
#include <stdio.h>
#define MAXLEN 80
#define MAXN 40
typedef struct studentRec{
  char lastname[MAXLEN];
  int mark;
}Student;
int main(){
  int total, i;
  Student stud_list[MAXN];
  printf("Co bao nhieu sinh vien? "); scanf("%d",&total);
  if(total > MAXN){
    printf("So qua lon! Khong du bo nho.\n"); exit(1);
  }
```

VÍ DỤ

```
printf("\nNhap danh sach ten va diem:\n");
  for(i=0;i<total;i++){
    printf("Sinh vien %d: ",i+1);
    scanf("%s %d",stud_list[i].name, &(stud_list[i].mark) );
  }
  printf("\nDanh sach nhung nguoi thi lai:\n");
  for(i=0;i<total;i++)
     if(stud_list[i].mark < 5){
        printf("Ten: %s\n",stud_list[i].name);
       printf("Diem: %d\n\n",stud_list[i].mark);
     }
  return 0;
}
```

CẤU TRÚC VÀ HÀM

- Giống như mọi biến khác, cấu trúc có thể được dùng làm tham số của hàm
- Cũng có hai cách truyền tham số cấu trúc cho hàm
  - Truyền cấu trúc theo dạng sử dụng giá trị của các trường, sẽ không làm thay đổi nội dung của biến cấu trúc gốc.
  - Truyền địa chỉ của cấu trúc để có thể làm thay đổi nội dung của cấu trúc gốc.

```
void readStudent(Student *s){
  printf("Please enter name and mark\n");
  scanf("%s",s->name);
  scanf("%f",&(s->mark));
}
int main(){
  Student studentA;
  readStudent(&studentA);
}
```

VÍ DỤ: Chương trình số phức

```c
#include <stdio.h>
typedef struct complexStruct{
  int real;
  int img;
}Complex;
Complex addComplex(Complex a, Complex b){
  Complex c;
  c.real = a.real + b.real;
  c.img = a.img + b.img;
  return c;
}
Complex readComplex(void){
  Complex c;
  printf("Nhap phan thuc ao cua so phuc: ");
  scanf("%d %d", &(c.real), &(c.img));
  return c;
}
```

```c
void printComplex(Complex c){
  if(c.img >= 0)
     printf("%d + %di", c.real, c.img);
  else
     printf("%d - %di", c.real, -c.img);
}
int main(){
  Complex a, b, tong;
  a = readComplex();
  b = readComplex();
  tong = addComplex(a, b);
  printf("Ket qua cua tong hai so phuc: ");
  printComplex(tong);
  return 0;
}
```

BÀI TẬP

- Một phân số được biểu diễn bằng một cấu trúc gồm hai trường tử số và mẫu số
- Viết hàm cho phép nhập giá trị cho một phân số
- Viết hàm hiển thị một phân số ra màn hình
- Xây dựng các hàm để tính một phân số rút gọn, tổng, hiệu, tích, thương của hai phân số
- Viết một chương trình để thử nghiệm các hàm đã xây dựng với hai phân số được nhập vào, sau đó tính tổng, hiệu, tích và thương của chúng.

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !