DỮ LIỆU CẤU TRÚC

NHẬP MÔN LẬP TRÌNH

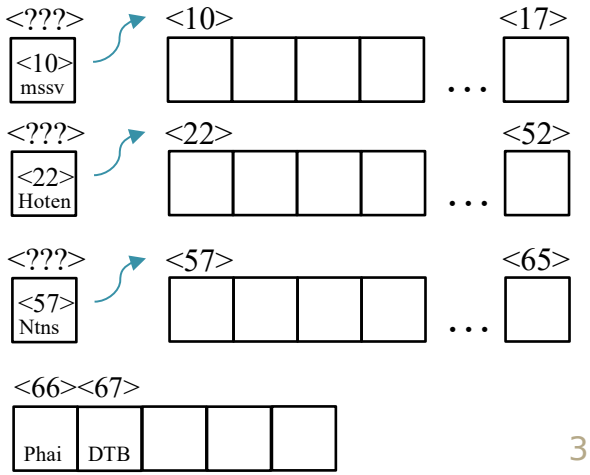
GVHD: Trương Toàn Thịnh

NỘI DUNG

- Giới thiệu
- Cú pháp khai báo/định nghĩa
- Bài tập

GIỚI THIỆU

- Quản lí **MỘT** sinh viên với các thông tin:
  - Mã số sinh viên
  - Họ tên
  - Ngày tháng năm sinh
  - Giới tính
  - Điểm trung bình
- Các biến đề xuất:
  - MSSV kiểu char[8]
  - HoTen kiểu char[31]
  - NTNS kiểu char[9]
  - Phai kiểu char
  - DTB kiểu float



GIỚI THIỆU

- Hàm nhập/xuất thông tin cho **MỘT** sinh viên

```
void Nhap(char* MSSV, char* HoTen,
   char* NTNS, char& Phai, float& DTB){
   …
}
void Xuat(char* MSSV, char* HoTen,
   char* NTNS, char Phai, float DTB){
   …
}
```

- Quản lí một lớp khoảng **100** sinh viên???

NỘI DUNG

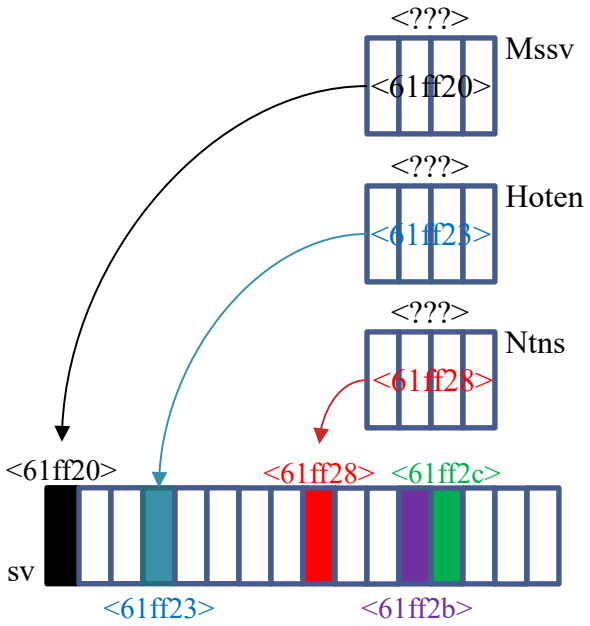
- Giới thiệu
- Cú pháp khai báo/định nghĩa
- Bài tập

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu cấu trúc SinhVien

Dòng	Mô tả
1	struct sinhvien{
2	char Mssv[3];
3	char HoTen[5];
4	char Ntns[3];
5	char Phai;
6	float Dtb;
7	};
8	typedef struct sinhvien SINHVIEN;

Ví dụ khai báo SINHVIEN sv

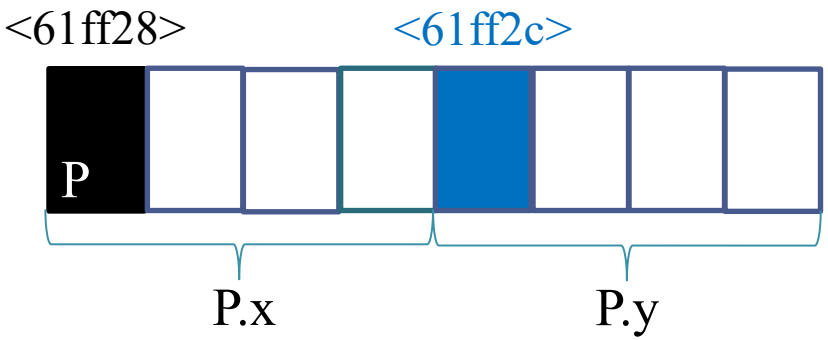


- Dòng mã số 8 sẽ thay thế cụm 'struct sinhvien' thành SINHVIEN

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu cấu trúc Point

	Mô tả
1	struct point{float x, y;};
2	typedef struct point POINT;
3	void main(){
4	POINT P = {2.0f, 3.0f}; // gán trực tiếp
5	cout << &P << ", " << &P.x << endl;
6	cout << &P.y << endl;
7	}

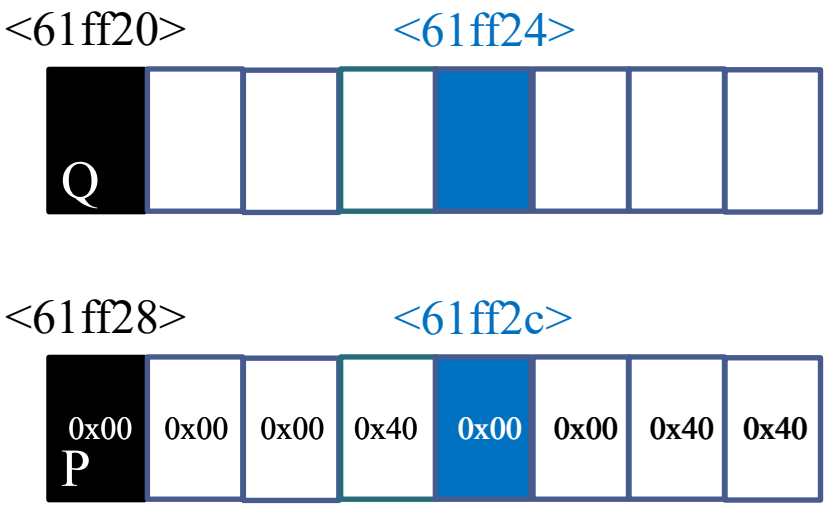


- Dòng mã số 4 sẽ gán trực tiếp 2 → P.x và 3 → P.y

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu cấu trúc Point

	Mô tả
1	struct point{float x, y;};
2	typedef struct point POINT;
3	void main(){
4	POINT P = {2.0f, 3.0f}, Q;
5	Q = P;
6	cout << &Q << ", " << &Q.x << endl;
7	cout << &Q.y << endl;
8	}

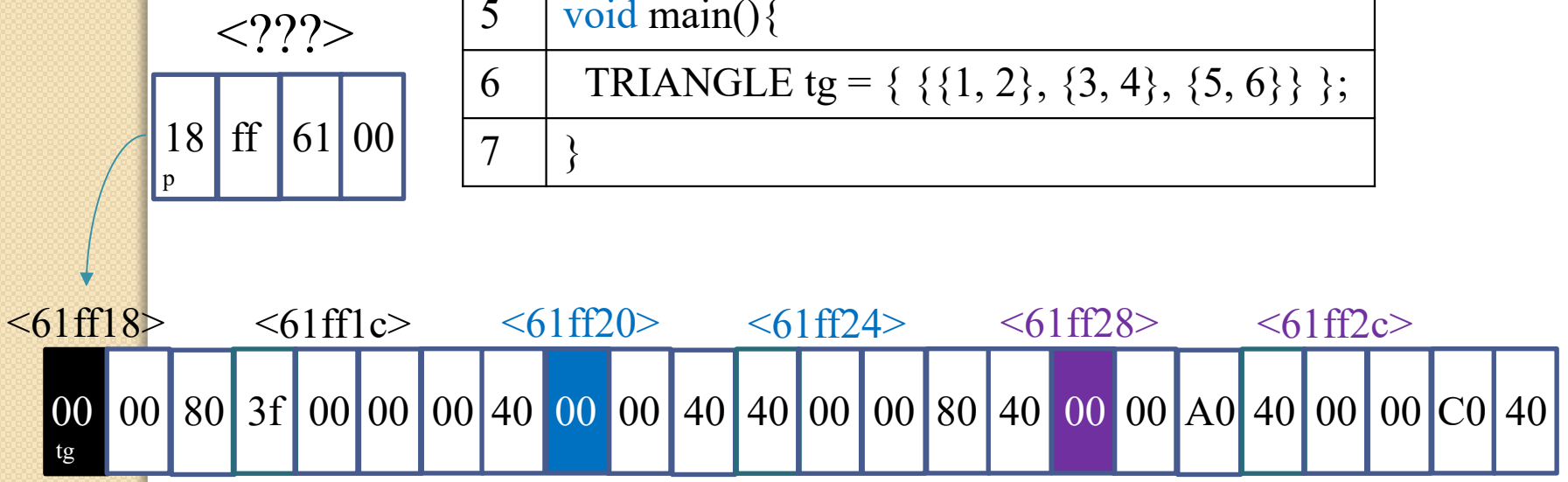


- Dòng 5 gán P.x → Q.x và P.y → Q.y
- Lưu ý: biểu diễn số 2.0 & 3.0 theo chuẩn IEEE754 Single precision 32-bit

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu cấu trúc Triangle

	Mô tả
1	struct point{float x, y;};
2	typedef struct point POINT;
3	struct triangle{ POINT p[3];};
4	typedef struct triangle TRIANGLE;
5	void main(){
6	TRIANGLE tg = { {{1, 2}, {3, 4}, {5, 6}} };
7	}



CÚ PHÁP KHAI BÁO

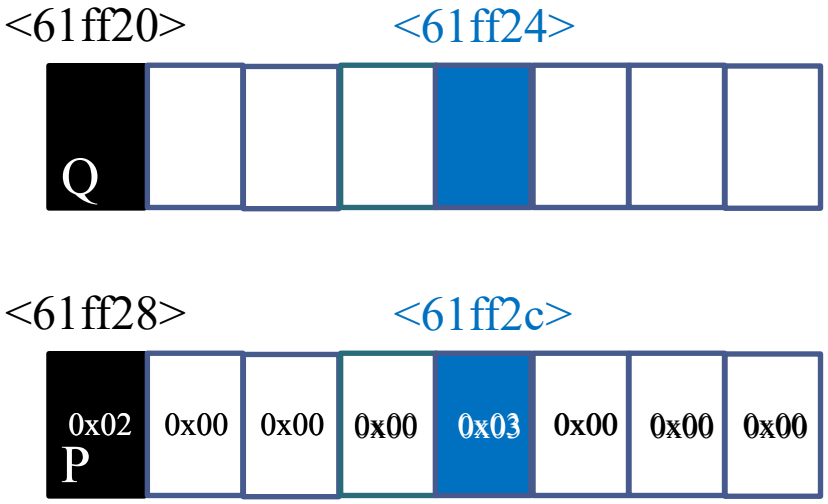
- Ví dụ: tìm trọng tâm tam giác

	Mô tả	
1	struct point{float x, y;};	void inputPoint(POINT& P){
2	typedef struct point POINT;	scanf("%d", &P.x);
3	struct triangle{ POINT p[3];};	scanf("%d", &P.y);
4	typedef struct triangle TRIANGLE;	}
5	void gravCenter(TRIANGLE t, POINT& c){	void inputTriangle(TRIANGLE& t){
6	c.x = (t.p[0].x + t.p[1].x + t.p[2].x)/3;	for(int i = 0; i < 3; i++)
7	c.y = (t.p[0].y + t.p[1].y + t.p[2].y)/3;	inputPoint(t.p[i]);
8	}	}
9	void main(){	void outputPoint(POINT P){
10	TRIANGLE tg; POINT M; inputTriangle(tg);	printf("(%d, ", P.x);
11	gravCenter(tg, M); outputPoint(M);	printf("%d)", P.y);
12	}	}

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu PhanSo

	Mô tả
1	struct phanso{long tu, mau;};
2	typedef struct phanso PHANSO;
3	void main(){
4	PHANSO P = {2, 3}, Q;
5	Q = P;
6	}



- Dòng 5 gán P.tu → Q.tu và P.mau → Q.mau
- Lưu ý: biểu diễn 2 & 3 theo dạng bù 2 (32-bit)

CÚ PHÁP KHAI BÁO

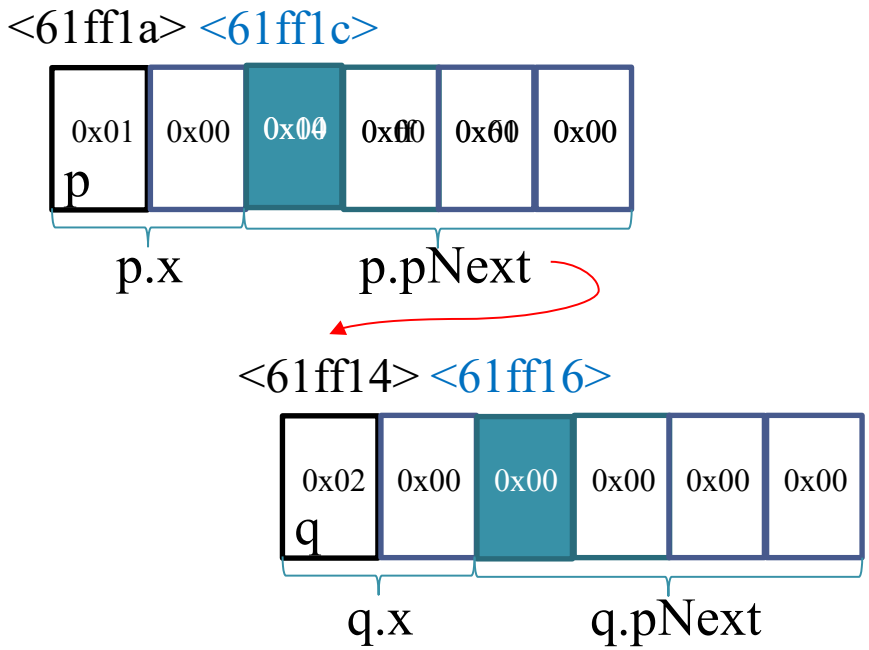
- Ví dụ: thao tác trên kiểu PHANSO

	Mô tả	
1	struct phanso{long tu, mau;};	void reduce(PHANSO& p){
2	typedef struct phanso PHANSO;	long gcd = **GCD**(p.tu, p.mau);
3	PHANSO add(PHANSO p, PHANSO q){	p.tu/=gcd;
4	PHANSO r;	}
5	r.tu = p.tu * q.mau + p.mau * q.tu;	void sub(PHANSO p, PHANSO q){
6	r.mau = t.mau * q.mau;	q.tu = -q.tu;
7	return r;	return add(p, q);
8	}	}
9	void main(){	void showFraction(PHANSO p){
10	PHANSO t = {1, 2}, s = {3, 4};	reduce(p);
11	showFraction(add(t, s)); showFraction(sub(t, s));	printf("%d/%d", p.tu, p.mau);
12	}	}

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu cấu trúc Node

	Mô tả
1	struct node{short x, struct node* pNext;};
2	typedef struct node NODE;
3	void main(){
4	NODE p = {1, NULL}, q = {2, NULL};
5	p.pNext = &q;
6	cout << p.pNext->val << endl;
7	}

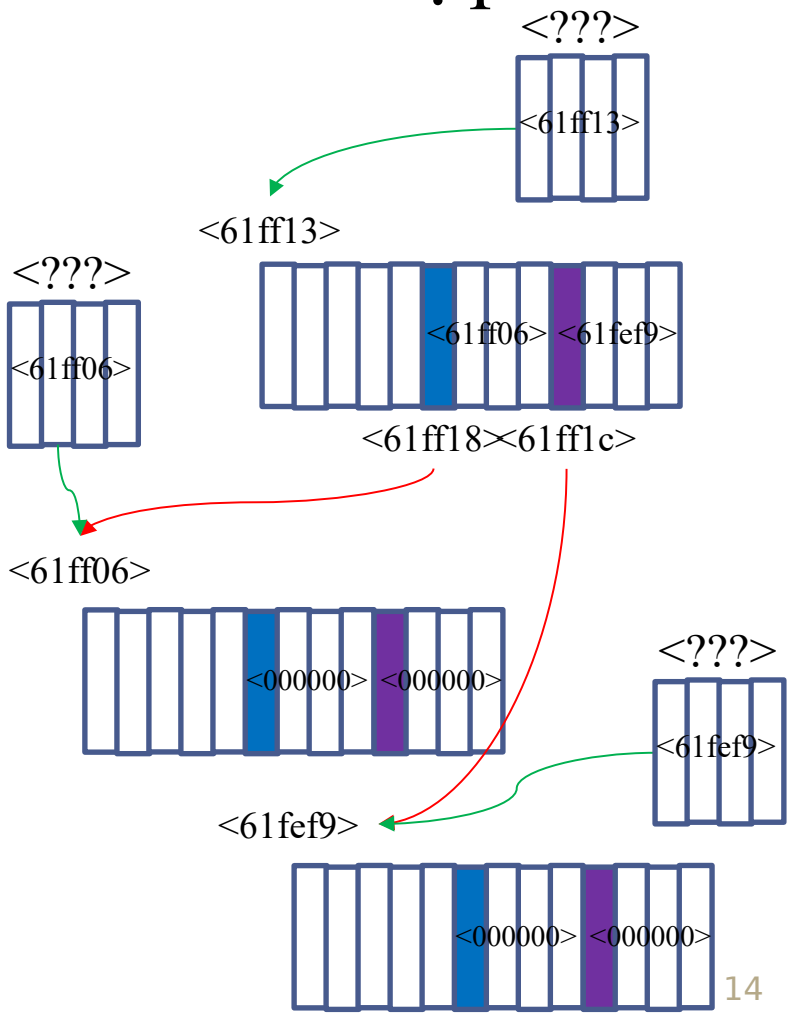


- Dòng số 5 gán địa chỉ của q.x → p.pNext

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu cấu trúc nhị phân

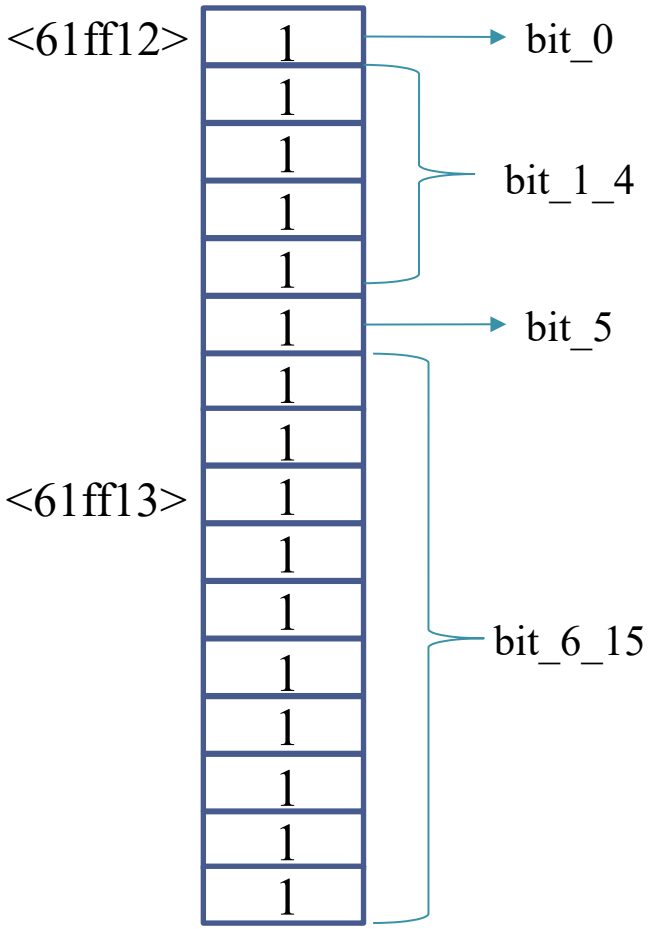
	Mô tả
1	struct person{
2	char name[5];
3	struct person* pa, *ma;
4	};
5	typedef struct person PERSON;
6	void main(){
7	PERSON p = {“Bob”, 0, 0};
8	PERSON q = {“Jack”, 0, 0}, t = {“Beth”, 0 ,0};
9	p.pa = &q; p.ma = &t;
10	}



CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu cấu trúc BIT

Dòng	Mô tả
1	struct bit_fields{
2	unsigned short bit_0: 1;
3	unsigned short bit_1_to_4: 4;
4	unsigned short bit_5: 5;
5	unsigned short bit_6_to_15: 10;
6	};
7	typedef struct bit_fields BF;
8	void main(){
9	BF b = {1, 15, 1, 1023};
10	}

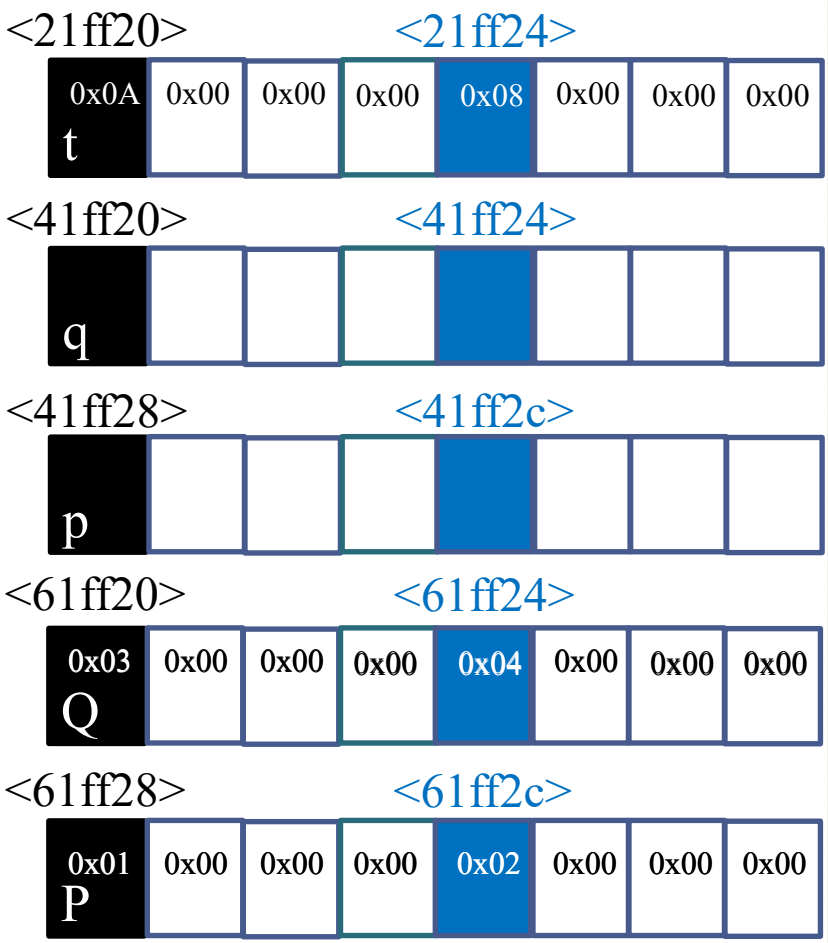


CÚ PHÁP KHAI BÁO

- Toán tử cho kiểu dữ liệu PhanSo

	Mô tả
1	struct phanso{long tu, mau;};
2	typedef struct phanso PHANSO;
3	PHANSO operator+(PHANSO p, PHANSO q){
4	PHANSO t;
5	t.tu = p.tu*q.mau + p.mau*q.tu;
6	t.mau = p.mau*q.mau;
7	return t;
8	}
9	PHANSO operator-(PHANSO p, PHANSO q){
10	q.mau = -q.mau;
11	return p + q;
12	}
13	void main(){
14	PHANSO P = {1, 2}, Q = {3, 4};
15	showFraction(P + Q);
16	}

Ví dụ: chạy hàm main

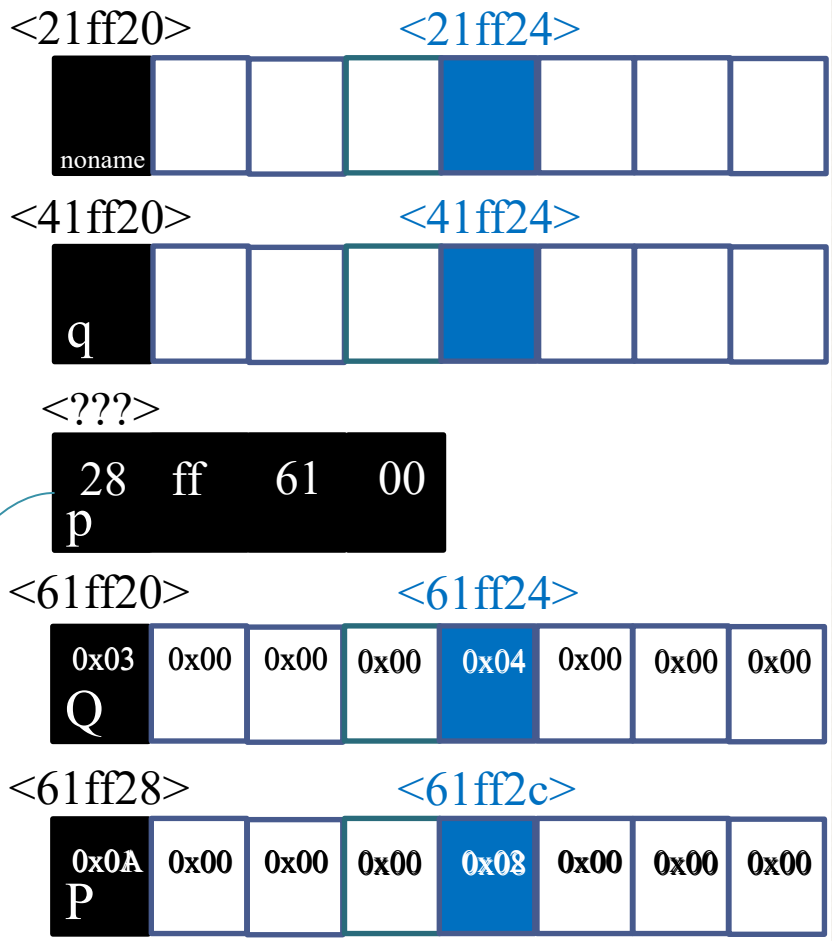


CÚ PHÁP KHAI BÁO

- Toán tử cho kiểu dữ liệu PhanSo

	Mô tả
1	struct phanso{long tu, mau;};
2	typedef struct phanso PHANSO;
3	PHANSO operator+=(PHANSO& p, PHANSO q){
4	p = p + q;
5	return p;
6	}
7	void main(){
8	PHANSO P = {1, 2}, Q = {3, 4};
9	showFraction(P += Q);
10	}

Ví dụ: chạy hàm main



CÚ PHÁP KHAI BÁO

- Xây dựng hàm với kiểu cấu trúc

Dòng	Mô tả
1	void Xuat(SINHVIEN sv){
2	printf("Ma so: %s", sv.Mssv);
3	printf("Ho ten: %s", sv.HoTen);
4	printf("Ngay sinh: %s", sv.Ntsn);
5	printf("Gioi tinh: %c", sv.Phai);
6	printf("Diem trung binh: %f", sv.Dtb);
7	}

CÚ PHÁP KHAI BÁO

- Xây dựng hàm với kiểu cấu trúc

Dòng	Mô tả
1	void Nhap(SINHVIEN& sv){
2	printf("Nhap ma so: ");
3	gets_s(sv.Mssv);
4	printf("Nhap ho ten: ");
5	gets_s(sv.Hoten);
6	printf("Nhap ngay sinh: ");
7	gets_s(sv.Ntns);
8	printf("Nhap gioi tinh: ");
9	scanf_s("%c", &sv.Phai);
10	printf("Nhap diem trung binh: ");
11	scanf_s("%f", &sv.Dtb);
12	}

CÚ PHÁP KHAI BÁO

- Sử dụng trong hàm main

Dòng	Mô tả
1	void main()
2	SINHVIEN sv;
3	Nhap(sv);
4	Xuat(sv);
5	}

- Cú pháp gán giá trị nhanh

Dòng	Mô tả
1	void main(){
2	SINHVIEN sv = {“0989821”, “Nguyen Van A”, “09/01/99”, ‘y’, 8};
3	Xuat(sv);
4	}

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu cấu trúc lồng

Dòng	Mô tả
1	struct Diem{
2	float x;
3	float y;
4	};
5	typedef struct Diem DIEM;
6	struct Tamgiac{
7	DIEM A, B, C;
8	}
9	typedef struct Tamgiac TAMGIAC;

CÚ PHÁP KHAI BÁO

- Xây dựng hàm với kiểu cấu trúc lồng

Dòng	Mô tả
1	void Xuat(DIEM d){
2	printf("(%f, %f)", d.x, d.y);
3	}
4	void Xuat(TAMGIAC tg){
5	Xuat(tg.A);
6	Xuat(tg.B);
7	Xuat(tg.C);
8	}

CÚ PHÁP KHAI BÁO

- Xây dựng hàm với kiểu cấu trúc lồng

Dòng	Mô tả
1	void Nhap(DIEM& d){
2	scanf("%f", &d.x);
3	scanf("%f", &d.y);
4	}
5	void Nhap(TAMGIAC& tg){
6	Nhap(tg.A);
7	Nhap(tg.B);
8	Nhap(tg.C);
9	}

CÚ PHÁP KHAI BÁO

- Sử dụng cấu trúc lồng trong hàm main

Dòng	Mô tả
1	void main()
2	TAMGIAC tg;
3	Nhap(tg);
4	Xuat(tg);
5	}

- Cú pháp gán giá trị nhanh

Dòng	Mô tả
1	void main(){
2	TAMGIAC tg = {{1, 2}, {2, 3}, {3, 4}};
3	Xuat(tg);
4	}

CÚ PHÁP KHAI BÁO

- Khai báo kiểu dữ liệu con trỏ cấu trúc

Dòng	Mô tả
1	void main(){
2	TAMGIAC* tg = new TAMGIAC;
3	*tg = {{1, 2}, {2, 3}, {3, 4}};
4	Xuat(**tg**);
5	}

- Do hàm "Xuat" cần một tham số kiểu TAMGIAC (KHÔNG phải kiểu TAMGIAC*) $\Rightarrow$ dùng toán tử "*" để truy xuất tới vùng nhớ 'thông thường'.

CÚ PHÁP KHAI BÁO

- Dùng toán tử '->' để truy xuất tới các thành phần kiểu con trỏ cấu trúc

Dòng	Mô tả
1	void main(){
2	TAMGIAC* tg = new TAMGIAC;
3	*tg = {{1, 2}, {2, 3}, {3, 4}};
4	Xuat(**tg**);
5	printf("%f", tg->A.x);}

- Truy xuất thành phần A của biến con trỏ cấu trúc tg bằng toán tử '->'.
- Truy xuất tiếp thành phần x của biến cấu trúc tg->A bằng toán tử '.'

CÚ PHÁP KHAI BÁO

- Khai báo mảng cấu trúc

Dòng	Mô tả
1	void Nhap(DIEM a[], int &n) {
2	printf("Nhap n: ");
3	scanf_s("%d", &n);
4	for (int i = 0;i < n; i++) Nhap(a[i]);
5	}
6	void Xuat(DIEM a[], int n) {
7	printf("n: %d\n", n);
8	for (int i = 0;i < n; i++) Xuat(a[i]);
9	}
10	void main() {
11	DIEM a[5]; int n;
12	Nhap(a, n); Xuat(a, n);}

CÚ PHÁP KHAI BÁO

- Gán nhanh giá trị mảng cấu trúc

Dòng	Mô tả
1	void main() {
2	DIEM a[5] = {{1, 2}, {2, 3}};
3	Xuat(a, 5);
4	}

- Với con trỏ cấu trúc ta cần viết tường minh từng phần tử

Dòng	Mô tả
1	void main() {
2	DIEM* a = new DIEM[2];
3	a[0] = {1,2}; a[1] = {3, 4};
4	}

NỘI DUNG

- Giới thiệu
- Cú pháp khai báo/định nghĩa
- Bài tập

BÀI TẬP

- Xây dựng kiểu PHANSO với các hàm:
  - +-×/
  - So sánh =, ≠, >, <, >=, <=
  - +=, -=
- Khai báo trong hàm main một mảng các PHANSO và thiết kế các hàm như sau
  - Tìm phân số lớn nhất trong mảng
  - Tính tổng các phân số trong mảng
  - Sắp xếp các phân số tăng dần