



ĐẠI HỌC KINH TẾ KỸ THUẬT CÔNG NGHIỆP
KHOA CÔNG NGHỆ THÔNG TIN

Bài Thực Tập 13

LẬP TRÌNH VỚI BIẾN CẤU TRÚC (tiếp)

NỘI DUNG

1

Mục tiêu bài học

2

Hướng dẫn học tập

3

Nội dung bài học

4

Giao nhiệm vụ tuần tiếp theo

MỤC TIÊU BÀI HỌC

- **Mục tiêu:**

Trang bị cho sinh viên kỹ năng lập trình cơ bản trong C++:

- + Khai báo và định nghĩa cấu trúc
- + Từ khoá typedef
- + Viết chương trình sử dụng cấu trúc
- + Phát hiện và sửa lỗi

- **Kết quả đạt được:**

Sinh viên thành thạo cách khai báo và định nghĩa kiểu dữ liệu cấu trúc, áp dụng giải các bài tập từ đơn giản đến phức tạp.

HƯỚNG DẪN HỌC TẬP

Để hoàn thành tốt bài học này sinh viên cần thực hiện những nhiệm vụ sau:

- Đọc trước tài liệu: “B13_Tailieu_TTLTCB” phần nhắc lại kiến thức lý thuyết mục “E. TÓM TẮT LÝ THUYẾT”
- Cài đặt, sử dụng được công cụ thực hành Cfree 5.0.
- Thực hành trên máy tính các bài thực hành mẫu.
- Hoàn thành các bài thực hành tự làm cuối bài học.
- Hoàn thành các bài tập giao về nhà.
- Trao đổi, thảo luận với giảng viên qua các phương thức:
 - + Thảo luận đặt câu hỏi trên diễn đàn.

NỘI DUNG BÀI HỌC

I. Hướng dẫn ban đầu (90 phút)

- 1.1 Tổng hợp lý thuyết
- 1.2 Hướng dẫn thực hành bài số 01
- 1.3 Hướng dẫn thực hành bài số 02

II. Hướng dẫn thường xuyên (90 phút)

- 2.1 Hướng dẫn thực hành bài số 03
- 2.2 Hướng dẫn thực hành bài số 04

III. Bài tập tự giải (90 phút)

I. HƯỚNG DẪN BAN ĐẦU

- Sinh viên cần cài đặt được công cụ thực hành.
- Yêu cầu sinh viên đọc lại và ghi nhớ các kiến thức lý thuyết trong mục E trong tài liệu “B13_*Tailieu_TTLTCB*”
- Hướng dẫn chi tiết ví dụ mẫu để hiểu rõ về cách lập trình với biến cấu trúc

1.1 TỔNG HỢP LÝ THUYẾT

❑ Một cấu trúc có thể lồng trong một cấu trúc khác. Tuy nhiên, một cấu trúc không thể lồng trong chính nó.

```
struct date
{
    int    day;
    int    month;
    int    year;
};

struct student
{
    int    no;
    char    name[20];
    struct date birthday;
    float    mark;
};
```

```
typedef struct
{
    int    day;
    int    month;
    int    year;
} date;

typedef struct
{
    int    no;
    char    name[20];
    date birthday;
    float    mark;
} student;
```

1.1 TỔNG HỢP LÝ THUYẾT

Truy cập vào thành phần của biến cấu trúc

- ❑ Các phần tử của cấu trúc được truy cập thông qua việc sử dụng toán tử chấm (.) hoặc (->), toán tử này còn được gọi là toán tử thành viên - membership.

- ❑ Cú pháp:

`tên_biến_cấu_trúc . tên_thành_phần`

`tên_con_trỏ_cấu_trúc -> tên_thành_phần`

- ❑ Ví dụ:

```
struct sanpham    sp1, *sp2, sp[10];  
cin.getline(sp1.tensp, 30);  
cin>> sp2 -> dongia;  
cin>>sp[9].soluong;
```


1.1 TỔNG HỢP LÝ THUYẾT

- ❑ Truy cập thành phần của cấu trúc lồng nhau.

```
typedef struct
{
    int    day;
    int    month;
    int    year;
} date;
typedef struct
{
    int    no;
    char   name[20];
    date birthday;
    float  mark;
} student;
```

```
student    sv1, dssv[50];
```

```
Truy cập trường năm sinh:
```

```
sv1.birthday.year
```

```
dssv[9].birthday.day
```

1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

Trong bài thực hành này sinh viên tìm hiểu và thực hành cách sử dụng biến cấu trúc.



Image Upload

Bài toán: Viết chương trình tạo một cấu trúc biểu diễn thông tin của một phân số gồm các trường: tử số, mẫu số. Tính tổng, tính hiệu, tính tích và tính thương của các phân số.

```

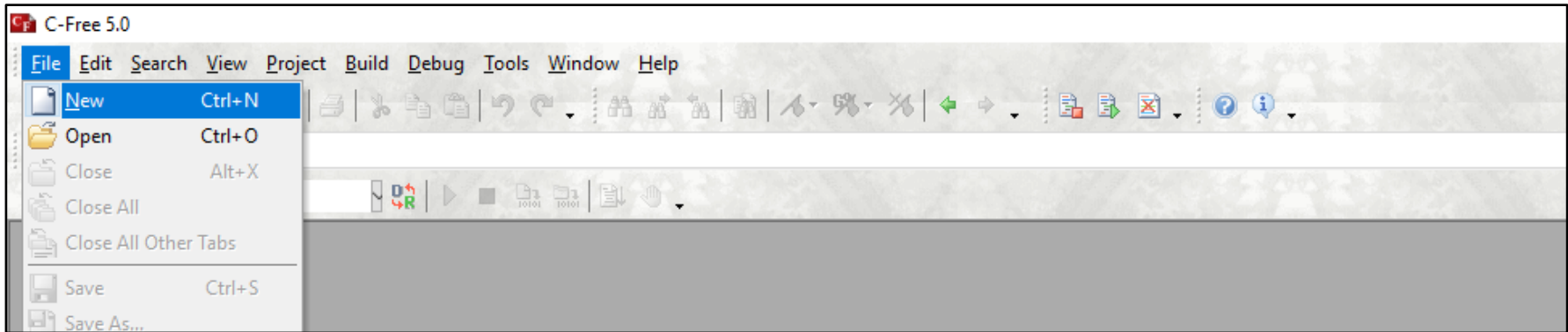
1  #include<iostream.h>
2  using namespace std;
3  typedef struct{
4      int tu;
5      int mau;
6  }Phan_so;
7  typedef struct{
8      Phan_so tong;
9      Phan_so hieu;
10     Phan_so tich;
11     Phan_so thuong;
12 }tinh;
13 void nhap(Phan_so a[],int n){
14     for(int i=0;i<n;i++){
15         cout<<"Nhap phan so thu "<<i+1<<":\n"
16             <<"\t Tu so = ";
17         cin>>a[i].tu;
18         cout<<"\t Mau so = ";
19         cin>>a[i].mau;
20     }
21 }
22 int ucln(int a, int b){
23     if(a<0)
24         a=a*-1;
25     if(b<0)
26         b*=-1;
27     while(a!= b){
28         if(a>b)
29             a-=b;
30         else
31             b-=a;
32     }
33     return a;
34 }
35 void rutgon(Phan_so &p) {
36     if(p.tu!=0){
37         int uc = ucln(p.tu,p.mau);
38         p.tu/= uc;
39         p.mau/= uc;
40     }
41 }
42 int main(){
43     Phan_so a[2];
44     nhap(a,2);
45     tinh s;
46     s.tong.tu = a[0].tu*a[1].mau + a[1].tu*a[0].mau;
47     s.tong.mau=a[0].mau*a[1].mau;
48     rutgon(s.tong);
49     s.hieu.tu = a[0].tu*a[1].mau - a[1].tu*a[0].mau;
50     s.hieu.mau=a[0].mau*a[1].mau;
51     rutgon(s.hieu);
52     s.tich.tu = a[0].tu*a[1].tu;
53     s.tich.mau=a[0].mau*a[1].mau;
54     rutgon(s.tich);
55     s.thuong.tu =a[0].tu*a[1].mau;
56     s.thuong.mau=a[0].mau*a[1].tu;
57     rutgon(s.thuong);
58     cout<<"\nTong = "<<s.tong.tu<<"/"<<s.tong.mau
59         <<"\nHieu = "<<s.hieu.tu<<"/"<<s.hieu.mau
60         <<"\nTich = "<<s.tich.tu<<"/"<<s.tich.mau<<endl;
61     <<"\nThuong = "<<s.thuong.tu<<"/"<<s.thuong.mau<<endl;

```

Image uploaded by Nguyễn Tuấn Tiến

1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

- **Bước 1:** Tạo mới một file*.cpp thực hiện thao tác File\New



- File mới xuất hiện, sinh viên chuyển sang bước 2 thực hiện gõ các câu lệnh theo các bước hướng dẫn.

1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

Bước 2: Khai báo thư viện cần dùng

```
#include <iostream.h>
```

```
#include <conio.h>
```

Bước 3: Khai báo (định nghĩa) cấu trúc:

```
typedef struct  
{   int      tu;  
    int      mau;  
}   phanso;
```

1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

Bước 4: Khai báo hàm `main()` là hàm chính của chương trình.

```
int main()
```

Bước 5: Gọi hàm xóa toàn bộ màn hình

```
clrscr();
```

1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

Bước 6: Nhập giá trị cho 2 phân số a và b

```
cout<<"\nNhap phan so a "<<endl;
cout<<"Tu so: "; cin>>a.tu;
cout<<"Mau so: "; cin>>a.mau;
cout<<"\nNhap phan so b "<<endl;
cout<<"Tu so: "; cin>>b.tu;
cout<<"Mau so: "; cin>>b.mau;
cout<<"Phan so thu nhat la:"<<a.tu<<"/"<<a.mau<<endl;
cout<<"Phan so thu hai la: "<<b.tu<<"/"<<b.mau<<endl;
```

1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

Bước 7: Tính tổng, hiệu hai phân số a và b:

```
c.tu = a.tu*b.mau + a.mau*b.tu;
```

```
c.mau = a.mau*b.mau;
```

```
cout<<"a + b = "<<(float)c.tu/c.mau<<endl;
```

```
c.tu = a.tu*b.mau - a.mau*b.tu;
```

```
c.mau = a.mau*b.mau;
```

```
cout<<"a - b = "<<(float)c.tu/c.mau<<endl;
```


1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

Bước 8: Tính tích, thương hai phân số

```
c.tu = a.tu*b.tu;
```

```
c.mau = a.mau*b.mau;
```

```
cout<<"a * b = "<<(float)c.tu/c.mau<<endl;
```

```
c.tu = a.tu*b.mau;
```

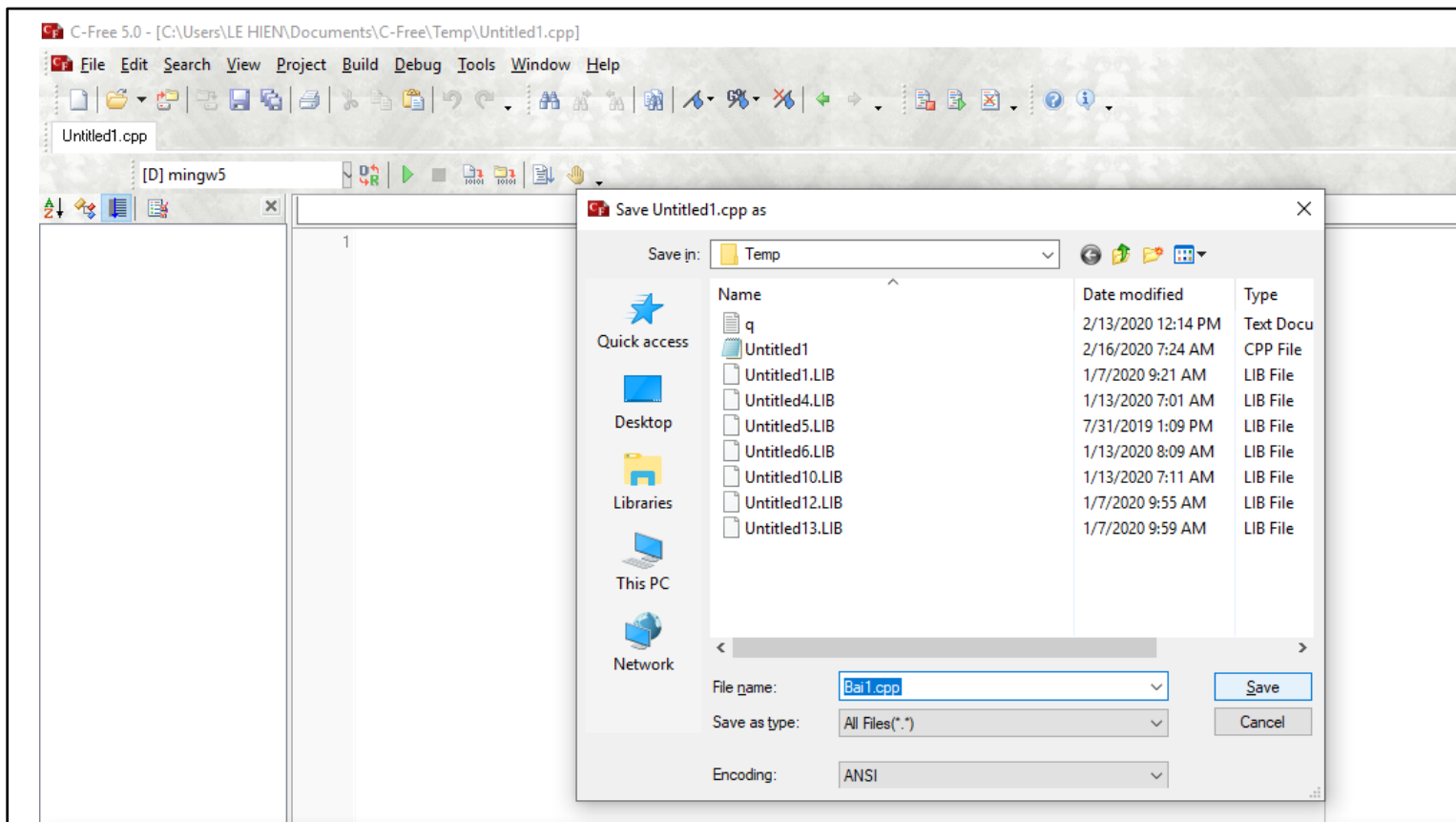
```
c.mau = a.mau*b.tu;
```

```
cout<<"a / b = "<<(float)c.tu/c.mau<<endl;
```

1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

- **Bước 9:** Lưu bài, chạy kiểm tra chương trình

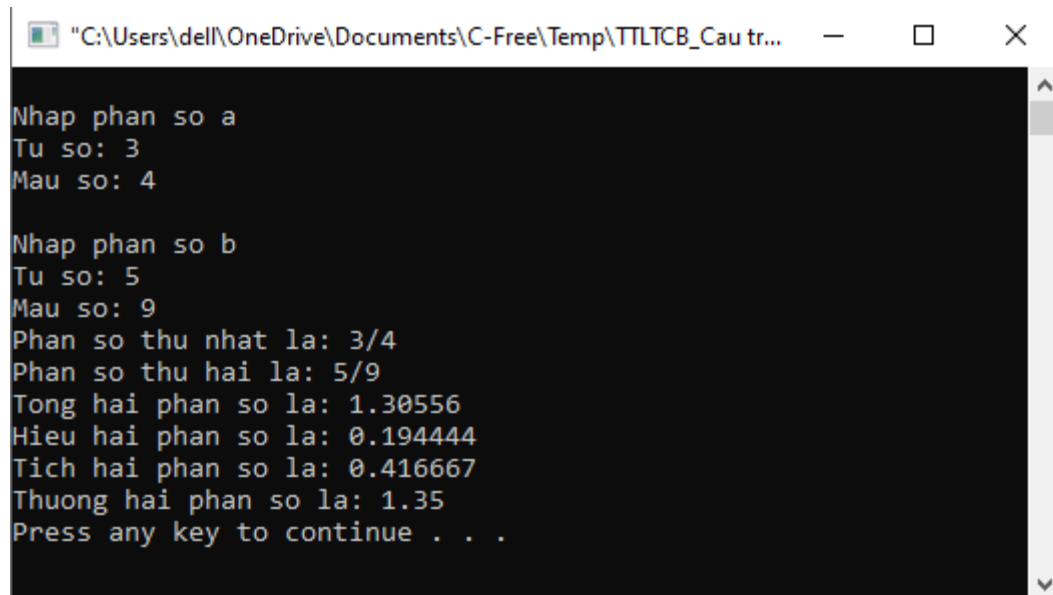
Sinh viên chọn trên thanh công cụ: File\Save\Gõ tên file cần lưu\Save



Ấn F5 để chạy và kiểm tra chương trình

1.2 HƯỚNG DẪN THỰC HÀNH BÀI 01

Lưu bài, biên dịch chương trình, nếu có lỗi phải sửa lỗi, nếu chương trình chạy đúng sẽ cho kết quả như sau:



```
"C:\Users\del\OneDrive\Documents\C-Free\Temp\TTLTCB_Cau tr...  —  □  ×  
Nhap phan so a  
Tu so: 3  
Mau so: 4  
  
Nhap phan so b  
Tu so: 5  
Mau so: 9  
Phan so thu nhat la: 3/4  
Phan so thu hai la: 5/9  
Tong hai phan so la: 1.30556  
Hieu hai phan so la: 0.194444  
Tich hai phan so la: 0.416667  
Thuong hai phan so la: 1.35  
Press any key to continue . . .
```

Tóm lại: Trong bài thực hành 01 chúng ta đã biết cách sử dụng biến cấu trúc C++, cách khai báo các biến, cách sử dụng câu lệnh nhập và xuất dữ liệu với biến cấu trúc.

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Trong bài thực hành này sinh viên tìm hiểu và thực hành cách sử dụng biến cấu trúc.



Image Upload

Bài toán:

Viết chương trình quản lý sinh viên bao gồm các thông tin, họ tên, ngày sinh, giới tính, điểm và thực hiện các yêu cầu sau:

Hàm nhập thông tin của một sinh viên sử dụng con trỏ cấu trúc làm đối số.

Hàm nhập thông tin danh sách sinh viên sử dụng con trỏ mảng làm tham số. Trong lời gọi sử dụng tên mảng để truyền cho hàm.

```

+ long castruc (Global S
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  struct sinhvien
7  {
8      string hoten;
9      int ngaysinh;
10     string gioitinh;
11     float diem;
12 };
13 typedef struct sinhvien SINHVIEN;
14 struct danhsach
15 {
16     SINHVIEN a[50];
17     int n;
18 };
19 typedef struct danhsach DANHSACH;
20 // hàm nhập thông tin sinh viên
21 void nhap_1_sv(SINHVIEN& x) {
22     SINHVIEN* p;
23     p = &x;
24     while (getchar() != '\n');
25     cout << "nhập họ tên sinh viên : ";
26     getline(cin, p->hoten);
27     cout << "nhập ngày sinh : ";
28     cin >> p->ngaysinh;
29     cin.ignore();
30     cout << "nhập giới tính sinh viên : ";
31     getline(cin, p->gioitinh);
32     cout << "nhập điểm số sinh viên : ";
33     cin >> p->diem;
34     cout << endl;
35 }
36 // hàm xuất sinh viên

```

Image uploaded by nguyên trung hiếu

```

1  #include<iostream.h>
2  #include<iomanip>
3  using namespace std;
4  typedef struct{
5      char ho_ten[30];
6      char ngay_sinh[15];
7      char gioi_tinh[10];
8      float diem;
9  }sinh_vien;
10 void nhap(sinh_vien *sv){
11     cout<<"Nhập họ và tên: ";
12     cin.ignore();
13     cin.getline(sv->ho_ten,30);
14     cout<<"Nhập ngày tháng năm sinh: ";
15     cin.getline(sv->ngay_sinh,30);
16     cout<<"Nhập giới tính: ";
17     cin.getline(sv->gioi_tinh,30);
18     cout<<"Nhập điểm: ";
19     cin>>sv->diem;
20 }
21 void xuat(sinh_vien *sv){
22     cout<<setw(30)<<sv->ho_ten
23         <<setw(15)<<sv->ngay_sinh
24         <<setw(10)<<sv->gioi_tinh
25         <<setw(10)<<sv->diem<<endl;
26 }
27 int main(){
28     int n;
29     cout<<"Nhập số lượng sinh viên: ";
30     cin>>n;
31     sinh_vien sv[n];
32     sinh_vien *svp=&sv[0];
33     for(int i=0;i<n;i++){
34         cout<<"\n\nNhập vào thông tin cho sinh viên thu "<<i+1<<endl;
35         nhap((svp+i));
36     }
37     cout<<"Danh sách sinh viên:\n"
38         <<setw(30)<<"Họ và tên"
39         <<setw(15)<<"Ngày sinh"
40         <<setw(10)<<"Giới tính"
41         <<setw(10)<<"Điểm"<<endl;
42     for(int i=0;i<n;i++){
43         xuat((svp+i));
44     }
45 }

```

Image uploaded by Nguyễn Tuấn Tiến

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Hàm in thông tin về một sinh viên sử dụng biến cấu trúc làm đối số



Image Upload

Hàm hiển thị danh sách sinh viên sử dụng con trỏ mảng làm tham số. Trong lời gọi sử dụng tên mảng để truyền cho hàm.

Hàm sửa thông tin về một sinh viên sử dụng tham chiếu cấu trúc làm đối số

Hàm sửa thông tin danh sách sinh viên sử dụng con trỏ mảng làm tham số. Trong lời gọi sử dụng tên mảng để truyền cho hàm.

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 1: Tạo file mới: tương tự như hướng dẫn ở bài thực hành 01

Bước 2: Khai báo thư viện cần dùng

```
#include <iostream.h>
```

```
#include <iomanip.h>
```


1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 3: Khai báo (định nghĩa) cấu trúc

```
typedef struct  
{  
    int    ng;  
    int    th;  
    int    nam;  
} ngaythang;
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 4: Khai báo (định nghĩa) cấu trúc

```
typedef struct
```

```
{
```

```
    char                hoten[30];
```

```
    ngaythang          ns;
```

```
    int                gt;
```

```
    float              diem;
```

```
} Sinhvien;
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 5: Hàm nhập thông tin về một sinh viên sử dụng con trỏ cấu trúc làm đối số

```
void  nhap(Sinhvien *p)
{
    cin.ignore();
    cout <<"\nHo ten:"; cin.getline(p->hoten, 30);
    cout <<"\nNgày sinh:";
    cin>>(p->ns).ng>>(p->ns).th>>(p->ns).nam;
    cout <<"\nGiới tính:"; cin>>p->gt ;
    cout <<"\nĐiểm: "; cin>>p->diem;
}
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 6: Hàm nhập thông tin danh sách sinh viên sử dụng con trỏ mảng làm tham số. Trong lời gọi sử dụng tên mảng để truyền cho hàm.

```
void nhapds(Sinhvien *a)
{
    int sosv = sizeof(lop) / sizeof(Sinhvien);
    for (int i=1; i<=sosv; i++)
        nhap(&a[i]) ;
}
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 7: Hàm in thông tin về một sinh viên sử dụng biến cấu trúc làm đối số

```
void in(Sinhvien x)
{
    cout << x.hoten << "\\t" ;
    cout << x.ns.ng << "/" << x.ns.th << "/" <<
    x.ns.nam << "\\t" ;
    cout << x.gt << "\\t";
    cout << x.diem << endl;
}
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 8: Hàm hiển thị danh sách sinh viên sử dụng con trỏ mảng làm tham số. Trong lời gọi sử dụng tên mảng để truyền cho hàm.

```
void      inds(const Sinhvien *a)
{
    int sosv = sizeof(lop) / sizeof(Sinhvien);
    for (int i=1; i<=sosv; i++)
        in(a[i]) ;
}
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 9: Hàm sửa thông tin về một sinh viên sử dụng tham chiếu cấu trúc làm đối số

```
void sua(Sinhvien &r)
{
    int chon;
    do {
        cout << "1: Sua ho ten" << endl;
        cout << "2: Sua ngay sinh" << endl;
        cout << "3: Sua gioi tinh" << endl;
        cout << "4: Sua diem" << endl;
        cout << "0: Thoi" << endl;
        cout << "Sua (0/1/2/3/4) ?"; cin>>chon;
        cin.ignore();
    }
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

```
switch (chon)
{
case 1: cin.getline(r.hoten,30) ; break;
case 2: cin >> r.ns.ng >> r.ns.th >> r.ns.nam ;
break;
case 3: cin >> r.gt ; break;
case 4: cin >> r.diem ; break;
} //end switch
} while (chon); //end do
}
```


1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 10: Hàm sửa thông tin danh sách sinh viên sử dụng con trỏ mảng làm tham số. Trong lời gọi sử dụng tên mảng để truyền cho hàm.

```
void suads(Sinhvien *a)
{
    int n;
    cout<<"Chon sinh vien can sua:"; cin>>n;
    cin.ignore();
    sua(a[n]);
}
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 11: Viết hàm main() là hàm chính của chương trình

```
int    main()  
{  
    nhapds(lop) ;  
    inds(lop) ;           //in ds vua nhap  
    suads(lop) ;  
    inds(lop) ;           //in ds vua sua  
    return 0;  
}
```

1.3 HƯỚNG DẪN THỰC HÀNH BÀI 02

Bước 12: Thực hiện tương tự như bài thực hành 01

Biên dịch chương trình, nếu có lỗi phải sửa lỗi, nếu chương trình chạy đúng sẽ cho kết quả như sau:

II. HƯỚNG DẪN THƯỜNG XUYỀN

- Sinh viên tiếp tục thực hành các bài tập theo sự gợi ý hướng dẫn.

Trong phần này yêu cầu:

- + Một số phần trong bài sinh viên phải tự thực hiện (Ví dụ: tạo file ban đầu, khai báo thư viện, những hàm có tính chất tương tự ở ví dụ trước).
- + Sinh viên phải tự hoàn thiện chương trình và chạy đúng.

2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

- **Bài toán:**

Viết chương trình khai báo cấu trúc số phức gồm 2 thành phần thực và ảo. Thực hiện cộng, trừ hai số phức.

2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

Hướng dẫn:

Định nghĩa cấu trúc sophuc

```
typedef struct  
{  
    float      thuc;  
    float      ao;  
} sophuc;
```

2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

Hàm cộng hai số phức

```
sophuc  cong(sophuc  p, sophuc  q)
{
    sophuc kq;
    kq.thuc = p.thuc + q.thuc ;
    kq.ao = p.ao + q.ao ;
    return kq;
}
```

2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

Hàm trừ hai số phức

```
sophuc tru(sophuc p, sophuc q)
{
    sophuc kq;
    kq.thuc = p.thuc - q.thuc ;
    kq.ao = p.ao - q.ao ;
    return kq;
}
```


2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

Đoạn lệnh in ra số phức

```
void In(sophuc a)
{
    cout << "(" << a.thuc << ", " << a.ao
    << ")" << endl ;
}
```

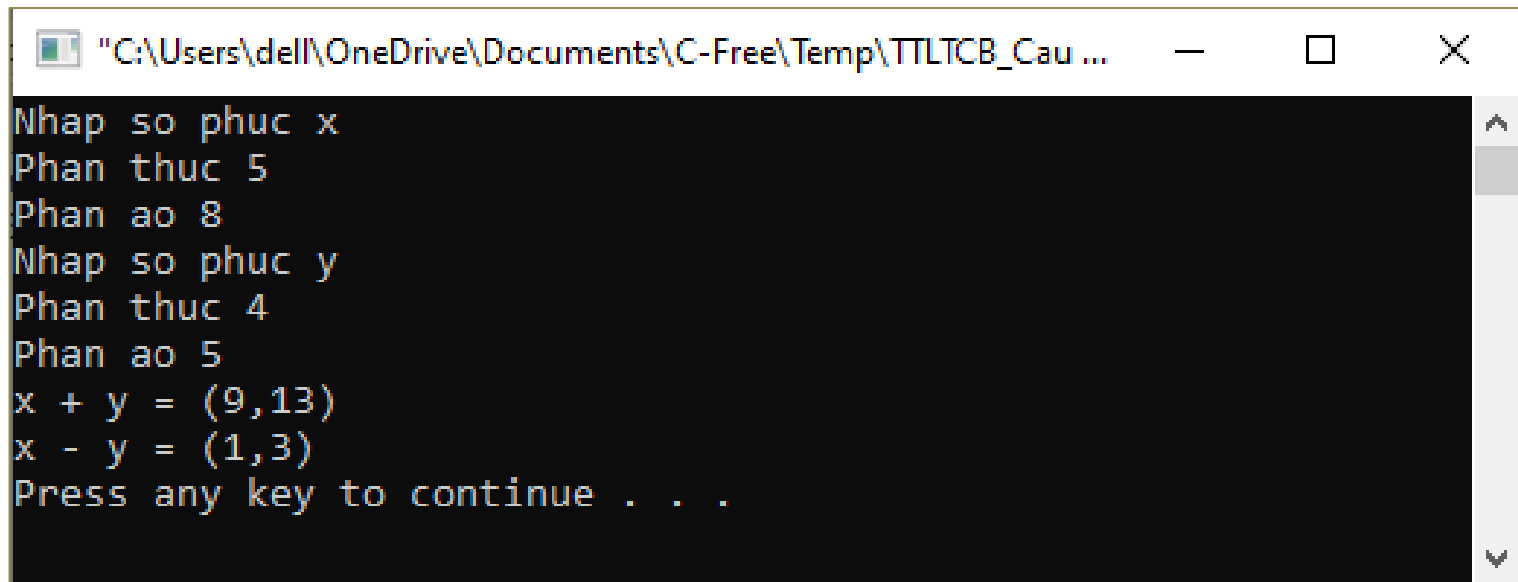
2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

Hàm main

```
int main()
{
    sophuc x, y;
    cout << "Nhap so phuc x "<<endl;
    cout<<"Phan thuc ";cin>>x.thuc;
    cout<<"Phan ao ";cin>>x.ao ;
    cout <<"Nhap so phuc y "<<endl;
    cout<<"Phan thuc ";cin>>y.thuc;
    cout<<"Phan ao ";cin>>y.ao;
    cout << "x + y = "; In(cong(x,y));
    cout << "x - y = "; In(tru(x,y));
    return 0;
}
```

2.1 HƯỚNG DẪN THỰC HÀNH BÀI 03

- Chương trình hoàn chỉnh sinh viên xem trong tài liệu “Tuan7_Tailieu_TTLTCB”
- Kết quả khi chạy chương trình



```
"C:\Users\dell\OneDrive\Documents\C-Free\Temp\TTLTCB_Cau ..."  
Nhap so phuc x  
Phan thuc 5  
Phan ao 8  
Nhap so phuc y  
Phan thuc 4  
Phan ao 5  
x + y = (9,13)  
x - y = (1,3)  
Press any key to continue . . .
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04



Image Upload

Bài toán:

Viết chương trình nhập và in thông tin về một lớp sinh viên.

- Nhập vào thông tin về điểm thi của các sinh viên trong lớp.
- Hiển thị sinh viên có điểm cao nhất lớp. (sử dụng con trỏ)

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 1: Tạo file mới: tương tự như hướng dẫn ở bài thực hành 01

Bước 2: Khai báo thư viện cần dùng

```
#include <iostream.h>
```

```
#include <iomanip.h>
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 3: Khai báo (định nghĩa) cấu trúc

```
typedef struct  
{  
    int    ng;  
    int    th;  
    int    nam;  
} ngaythang;
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 4: Khai báo (định nghĩa) cấu trúc

```
typedef struct
```

```
{
```

```
    char                hoten[30];
```

```
    ngaythang          ns;
```

```
    int                gt;
```

```
    float              diem;
```

```
} Sinhvien;
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 5: Hàm nhập thông tin về một sinh viên sử dụng con trỏ cấu trúc làm đối số

```
void  nhap(Sinhvien *p)
{
    cin.ignore();
    cout <<"\nHo ten:"; cin.getline(p->hoten, 30);
    cout <<"\nNgày sinh:";
    cin>>(p->ns).ng>>(p->ns).th>>(p->ns).nam;
    cout <<"\nGiới tính:"; cin>>p->gt ;
    cout <<"\nĐiểm: "; cin>>p->diem;
}
```


2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 6: Hàm nhập thông tin danh sách sinh viên sử dụng con trỏ mảng làm tham số. Trong lời gọi sử dụng tên mảng để truyền cho hàm.

```
void nhapds(Sinhvien *a)
{
    int sosv = sizeof(lop) / sizeof(Sinhvien);
    for (int i=1; i<=sosv; i++)
        nhap(&a[i]) ;
}
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 7: Hàm in thông tin về một sinh viên sử dụng biến cấu trúc làm đối số

```
void in(Sinhvien x)
{
    cout << x.hoten << "\\t" ;
    cout << x.ns.ng << "/" << x.ns.th << "/" <<
    x.ns.nam << "\\t" ;
    cout << x.gt << "\\t";
    cout << x.diem << endl;
}
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 8: Hàm hiển thị danh sách sinh viên sử dụng con trỏ mảng làm tham số. Trong lời gọi sử dụng tên mảng để truyền cho hàm.

```
void        inds(const Sinhvien *a)
{
    int sosv = sizeof(lop) / sizeof(Sinhvien);
    for (int i=1; i<=sosv; i++)
        in(a[i]) ;
}
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 9: Hàm in sinh viên có điểm cao nhất

```
void insvdiemcaonhat(Sinhvien *a)
{
    float max;
    int sosv = sizeof(lop) / sizeof(Sinhvien);
    max = a[0].diem;
    for (int i=1; i<=sosv; i++)
        if (max < a[i].diem)
            max = a[i].diem;
    int j = 0;
    cout << "Diem cao nhat lop:" << endl;
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
for (int i = 1; i <= sosv; i++)  
{  
    if (a[i].diem == max)  
    {  
        cout<<"#"<<j + 1<<endl;  
        cout<<a[i].hoten<<" "<<a[i].ns.ng  
        cout<<"/"<<a[i].ns.th<<"/";  
        cout<<a[i].ns.nam<<" ";  
        cout<<a[i].gt<<" "<<a[i].diem<<endl;  
        j++;  
    }  
}  
}
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 11: Viết hàm main() là hàm chính của chương trình

```
int main()  
{  
    nhapds(lop);  
    inds(lop);           //in ds vua nhap  
    insvdiemcaonhat(lop);  
    return 0;  
}
```

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

Bước 12: Thực hiện tương tự như bài thực hành 01

Biên dịch chương trình, nếu có lỗi phải sửa lỗi, nếu chương trình chạy đúng sẽ cho kết quả như sau:

2.2 HƯỚNG DẪN THỰC HÀNH BÀI 04

```
"C:\Users\dell\OneDrive\Documents\C-Free\Temp\TLTCB_Cau truc(tiep)..."
Nhap danh sach hoc sinh
Ho ten: Nga
Ngay sinh: 4 8 2003
Gioi tinh: 0
Diem: 7
Ho ten: Tung
Ngay sinh: 12 7 2002
Gioi tinh: 1
Diem: 6
Ho ten: Van
Ngay sinh: 4 10 2003
Gioi tinh: 0
Diem: 8
ga      4/8/2003      0      7
Tung    12/7/2002    1      6
Van     4/10/2003    0      8
Diem cao nhat lop:
#1
Van 4/10/2003 0 8
Press any key to continue . . .
```


III. BÀI TẬP TỰ GIẢI

Bài 5: Viết chương trình thực hiện phân tích thống kê một lớp học khoảng 20 sinh viên. Thông tin của mỗi sinh viên bao gồm ID, tên, tuổi, điểm tổng kết học kì 1, điểm tổng kết học kì 2. Những thông tin cần thống kê bao gồm:

- a) Điểm trung bình cuối năm của cả lớp.
- b) Điểm tổng kết cuối năm của sinh viên nào là cao nhất.
- c) Liệt kê danh sách những sinh viên có tiến bộ trong học tập (điểm tổng kết học kì 2 cao hơn điểm tổng kết học kì 1).

III. BÀI TẬP TỰ GIẢI

Bài 6: Viết chương trình tạo struct công nhân gồm các trường: họ tên, năm sinh và giờ làm.

- a) Nhập một số n , tạo và nhập một mảng n công nhân.
- b) Liệt kê tên các công nhân nhận thưởng biết rằng công nhân là các công nhân có giờ làm > 40 giờ.

III. BÀI TẬP TỰ GIẢI

Bài 7: Cho một danh sách n sinh viên. Thông tin về mỗi sinh viên gồm: mã sv, họ tên, năm sinh, điểm trung bình. Viết chương trình thực hiện các công việc sau:

- a) Khai báo dữ liệu kiểu cấu trúc lưu trữ thông tin cho một sinh viên.
- b) Nhập vào từ bàn phím một danh sách sinh viên.
- c) Sắp xếp danh sách sinh viên theo thứ tự giảm dần của tuổi, in ra danh sách sau khi đã sắp xếp (với đầy đủ thông tin dưới dạng bảng).

III. BÀI TẬP TỰ GIẢI

Bài 8: Viết chương trình khai báo kiểu dữ liệu để biểu diễn một điểm trong hệ tọa độ Oxy. Hãy viết hàm thực hiện các công việc sau:

- a) Tìm những điểm đối xứng của nó qua tung độ, hoành độ, tọa độ tâm.
- b) Hãy tính tổng, hiệu, tích của hai điểm trong mặt phẳng tọa độ Oxy.

III. BÀI TẬP TỰ GIẢI

Bài 9: Viết chương trình khai báo kiểu dữ liệu để biểu diễn một phân số. Hãy viết hàm thực hiện những công việc sau:

- a) Rút gọn phân số.
- b) So sánh hai phân số.

Bài 10: Để quản lý các mặt hàng của một siêu thị, mỗi một mặt hàng được lưu trữ các thông tin: mã hàng, tên hàng, số lượng, đơn giá, thành tiền. Hãy lập chương trình thực hiện các yêu cầu sau:

- a) Khai báo kiểu dữ liệu cấu trúc lưu trữ thông tin cho một mặt hàng.
- b) Nhập vào từ bàn phím một danh sách n mặt hàng.
- c) Tính thành tiền cho các mặt hàng biết: thành tiền = số lượng * đơn giá.

IV. TỔNG KẾT

Kiến thức cần ghi nhớ trong buổi thực hành:

- 1 Khai báo cấu trúc
- 2 Từ khoá typedef
- 3 Khai báo biến cấu trúc
- 4 Viết chương trình sử dụng biến cấu trúc

BÀI TẬP VỀ NHÀ

Sinh viên làm tất cả các bài tập trong phần:

**“F. BÀI TẬP THỰC HÀNH TRÊN PHÒNG MÁY” trong
B13_Tailieu_TTLTCB**

Lời ngỏ

Trong quá trình học tập nếu sinh viên không hiểu phần nào thì liên hệ trao đổi với giảng viên qua hình thức gửi câu hỏi trên diễn đàn hoặc gửi vào email giảng viên.

V. GIAO NHIỆM VỤ TUẦN TIẾP THEO

1. Hoàn thành tất cả các bài thực hành trong buổi học và bài tập trong mục D file “B13_Baitap_TTLTCB.docx”.
2. Sinh viên đọc tài liệu và chuẩn bị trước nội dung học của tuần tiếp theo:

Lập trình với biến tệp

- + Đọc tệp văn bản
- + Ghi tệp văn bản
- + Đọc tệp nhị phân
- + Ghi tệp nhị phân
- + Viết chương trình sử dụng tệp
- + Phát hiện và sửa lỗi