

# Chương 7: Giới thiệu tổng quan về lập trình

## Phần a: Dữ liệu có cấu trúc-Bài tập

### **Nhập môn lập trình**

Trình bày: Nguyễn Sơn Hoàng Quốc

Email: [nshquoc@fit.hcmus.edu.vn](mailto:nshquoc@fit.hcmus.edu.vn)

# Câu 1 – Khoảng cách gốc tọa độ

- Viết hàm xác định khoảng cách giữa một điểm so với gốc tọa độ.

cuu duong than cong . com

cuu duong than cong . com

# Câu 1 – Khoảng cách gốc tọa độ

```
1. float TinhKhoangCach(DIEM P)
2. {
3.     float khoangCach;
4.     cuu duong than cong . com
5.     khoangCach = sqrt(P.x * P.x + P.y * P.y);
6.
7.     return khoangCach;
8. }
```

## Câu 2 – Chu vi tam giác

- Viết chương trình nhập vào một tam giác gồm 3 đỉnh. Tính chu vi của tam giác đó và xuất kết quả.

cuu duong than cong . com

cuu duong than cong . com

## Câu 2 – Chu vi tam giác

```
1. #include <iostream>
2. using namespace std;
3. struct DIEM
4. {
5.     float x;
6.     float y;
7. };
8. struct TAM_GIAC
9. {
10.    DIEM A;
11.    DIEM B;
12.    DIEM C;
13.};
```

## Câu 2 – Chu vi tam giác

1. `void` Nhap(`DIEM&` diem);
2. `void` Nhap(`TAM_GIAC&` tg);
3. `float` TinhKhoangCach(`DIEM` A, `DIEM` B);
4. `float` TinhChuVi(`TAM_GIAC` tg);

cuu duong than cong . com

## Câu 2 – Chu vi tam giác

```
1. void main()  
2. {  
3.     TAM_GIAC tg;  
4.     float chuVi;  
5.  
6.     cout << "Nhap tam giac" << endl;  
7.     Nhap(tg);  
8.  
9.     chuVi = TinhChuVi(tg);  
10.  
11.    cout << "Chu vi" << chuVi;  
12.}
```

## Câu 2 – Chu vi tam giác

```
1. void Nhap(DIEM& D)
2. {
3.     cout << "Nhap hoành do = ";
4.     cin >> D.x;
5.     cout << "Nhap tung do = ";
6.     cin >> D.y;
7. }
```

cuu duong than cong . com



## Câu 2 – Chu vi tam giác

```
1. void Nhap(TAM_GIAC& tg)
2. {
3.     cout << "Nhap dinh A = " << endl;
4.     Nhap(tg.A);
5.     cout << "Nhap dinh B = " << endl;
6.     Nhap(tg.B);
7.     cout << "Nhap dinh C = " << endl;
8.     Nhap(tg.C);
9. }
```

## Câu 2 – Chu vi tam giác

```
1. float TinhKhoangCach(DIEM A, DIEM B)
2. {
3.     float khoangCach;
4.
5.     khoangCach = sqrt((A.x-B.x)*(A.x-B.x) +
6.                        (A.y-B.y)*(A.y-B.y));
7.     return khoangCach;
8. }
```

## Câu 2 – Chu vi tam giác

```
1. float TinhChuVi(TAM_GIAC tg)
2. {
3.     float chuVi, a, b, c;
4.
5.     a = TinhKhoangCach(tg.B, tg.C);
6.     b = TinhKhoangCach(tg.A, tg.C);
7.     c = TinhKhoangCach(tg.A, tg.B);
8.     chuVi = a + b + c;
9.
10.    return chuVi;
11.}
```

## Câu 3 – Hàm hiệu hai phân số

- Viết hàm tính hiệu hai phân số

cuu duong than cong . com

cuu duong than cong . com

## Câu 3 – Hàm hiệu hai phân số

1. struct PHAN\_SO

2. {

3.   int tu;

4.   int mau;

5. };

cuu duong than cong . com

## Câu 3 – Hàm hiệu hai phân số

```
1. PHAN_SO TinhHieu(PHAN_SO A, PHAN_SO B)
2. {
3.     PHAN_SO psHieu;
4.     cuu duong than cong . com
5.     psHieu.tu=A.tu*B.mau-B.tu*A.mau;
6.     psHieu.mau=A.mau*B.mau;
7.
8.     return psHieu;
9. }
```

## Câu 4 – Hàm so sánh hai phân số

- Viết hàm so sánh hai phân số (hàm trả về một trong 3 giá trị -1, 0, và 1)

cuu duong than cong . com

cuu duong than cong . com

## Câu 4 – Hàm so sánh hai phân số

1. struct PHAN\_SO

2. {

3.   int tu;

4.   int mau;

5. };

cuu duong than cong . com



## Câu 4 – Hàm so sánh hai phân số

```
1. int SoSanh(PHAN_SO A, PHAN_SO B)
2. {
3.     int ketQua = 0; // giả sử bằng nhau
4.     float giaTri1, giaTri2;

5.     giaTri1 = (float)A.tu / A.mau;
6.     giaTri2 = (float)B.tu / B.mau;
7.     if (giaTri1 > giaTri2)
8.         ketQua = 1;
9.     if (giaTri1 < giaTri2)
10.        ketQua = -1;

11.    return ketQua;
12.}
```

## Câu 5 – Tìm điểm đối xứng Oxyz

- Viết hàm tìm điểm đối xứng qua gốc tọa độ  $Oxyz$  và xuất kết quả.

cuu duong than cong . com

cuu duong than cong . com

## Câu 5 – Tìm điểm đối xứng Oxyz

- Viết hàm tìm điểm đối xứng qua gốc tọa độ  $Oxyz$  và xuất kết quả.

cuu duong than cong . com

cuu duong than cong . com

## Câu 5 – Tìm điểm đối xứng Oxyz

```
1. struct DIEM_KHONG_GIAN
2. {
3.     float x;
4.     float y;
5.     float z;
6. };
```

cuu duong than cong . com

## Câu 5 – Tìm điểm đối xứng Oxyz

```
1. DIEM_KHONG_GIAN TimDoiXung(DIEM_KHONG_GIAN D)
2. {
3.     DIEM_KHONG_GIAN kq;
4.
5.     kq.x = D.x;
6.     kq.y = D.y;
7.     kq.z = D.z;
8.
9.     return kq;
10.}
```

## Câu 6 – Tam giác hợp lệ, tính diện tích

- Viết chương trình nhập vào ba điểm. Kiểm tra ba điểm có tạo thành một tam giác không. Nếu có tính diện tích tam giác và xuất kết quả.

cuu duong than cong . com

## Câu 6 – Tam giác hợp lệ, tính diện tích

```
1. #include <iostream>
```

```
2. using namespace std;
```

```
3. struct DIEM
```

```
4. {
```

```
5.     float x;
```

```
6.     float y;
```

```
7. };
```

```
8. struct TAM_GIAC
```

```
9. {
```

```
10.     DIEM A;
```

```
11.     DIEM B;
```

```
12.     DIEM C;
```

```
13. };
```

## Câu 6 – Tam giác hợp lệ, tính diện tích

1. `void Nhap(DIEM& diem);`
2. `void Nhap(TAM_GIAC& tg);`
3. `bool KiemTraHopLe(TAM_GIAC tg);`
4. `float TinhKhoangCach(DIEM A, DIEM B);`
5. `float TinhDienTich(TAM_GIAC tg);`

cuu duong than cong . com



## Câu 6 – Tam giác hợp lệ, tính diện tích

```
1. void main()
2. {
3.     TAM_GIAC tg;
4.     float dienTich;
5.
6.     cout << "Nhap tam giac" << endl;
7.     Nhap(tg);
8.
9.     if (KiemTraHopLe(tg))
10.    {
11.        dienTich = TinhDienTich(tg);
12.        cout << "Dien tich " << dienTich;
13.    }
14.}
```

## Câu 6 – Tam giác hợp lệ, tính diện tích

```
1. void Nhap(DIEM& D)
2. {
3.     cout << "Nhap hoành do = ";
4.     cin >> D.x;
5.     cout << "Nhap tung do = ";
6.     cin >> D.y;
7. }
```

cuu duong than cong . com

## Câu 6 – Tam giác hợp lệ, tính diện tích

```
1. void Nhap(TAM_GIAC& tg)
2. {
3.     cout << "Nhap dinh A = " << endl;
4.     Nhap(tg.A);
5.     cout << "Nhap dinh B = " << endl;
6.     Nhap(tg.B);
7.     cout << "Nhap dinh C = " << endl;
8.     Nhap(tg.C);
9. }
```

## Câu 6 – Tam giác hợp lệ, tính diện tích

```
1. float TinhKhoangCach(DIEM A, DIEM B)
2. {
3.     float khoangCach;
4.
5.     khoangCach=sqrt( (A.x-B.x)*(A.x-B.x)+
6.                     (A.y-B.y)*(A.y-B.y));
7. return khoangCach;
8. }
```

## Câu 6 – Tam giác hợp lệ, tính diện tích

```
1. float TinhDienTich(TAM_GIAC tg)
2. {
3.     float dienTich, p, a, b, c;
4.
5.     a = TinhKhoangCach(tg.B, tg.C);
6.     b = TinhKhoangCach(tg.A, tg.C);
7.     c = TinhKhoangCach(tg.A, tg.B);
8.
9.     p = (a + b + c)/2;
10.
11.     dienTich=sqrt(p*(p-a)*(p-b)*(p-c));
12.
13.     return dienTich;
14. }
```

## Câu 6 – Tam giác hợp lệ, tính diện tích

```
1. bool KiemTraHopLe(TAM_GIAC tg)
2. {
3.     bool hopLe = false; // mặc định ko hợp lệ
4.     float a, b, c;
5.
6.     a = TinhKhoangCach(tg.B, tg.C);
7.     b = TinhKhoangCach(tg.A, tg.C);
8.     c = TinhKhoangCach(tg.A, tg.B);
9.
10.    if ((a<b+c) && (b<a+c) && (c<a+b))
11.        hopLe = true;
12.
13.    return hopLe;
14.}
```

## Câu 7 – Vị trí tương đối hình tròn

- Viết chương trình nhập vào hai đường tròn. Xuất ra vị trí tương đối của hai hình tròn (không cắt nhau, cắt nhau, tiếp xúc).

cuu duong than cong . com

cuu duong than cong . com

## Câu 7 – Vị trí tương đối hình tròn

```
1. #include <iostream>
```

```
2. using namespace std;
```

```
3. struct DIEM
```

```
4. {
```

```
5.     float x;
```

```
6.     float y;
```

```
7. };
```

```
8. struct DUONG_TRON
```

```
9. {
```

```
10.     DIEM O;
```

```
11.     float R;
```

```
12. };
```



## Câu 7 – Vị trí tương đối hình tròn

1. `void Nhap(DIEM& diem);`
2. `void Nhap(DUONG_TRON& tg);`
3. `float TinhKhoangCach(DIEM A, DIEM B);`
4. `void TuongDoi(DUONG_TRON dt1, DUONG_TRON dt2);`

cuu duong than cong . com

## Câu 7 – Vị trí tương đối hình tròn

```
1. void main()
2. {
3.     DUONG_TRON dt1, dt2;
4.
5.     cout << "Nhap duong tron 1" << endl;
6.     Nhap(dt1);
7.
8.     cout << "Nhap duong tron 2" << endl;
9.     Nhap(dt2);
10.
11.     TuongDoi(dt1, dt2);
12. }
```

## Câu 7 – Vị trí tương đối hình tròn

```
1. void Nhap(DIEM& D)
2. {
3.     cout << "Nhap hoành do = ";
4.     cin >> D.x;
5.     cout << "Nhap tung do = ";
6.     cin >> D.y;
7. }
```

cuu duong than cong . com

## Câu 7 – Vị trí tương đối hình tròn

```
1. void Nhap(DUONG_TRON& dt)
2. {
3.     cout << "Nhap tam = " << endl;
4.     Nhap(dt.O);
5.     cout << "Nhap ban kinh = ";
6.     cin >> dt.R;
7. }
```

cuu duong than cong . com

## Câu 7 – Vị trí tương đối hình tròn

```
1. float TinhKhoangCach(DIEM A, DIEM B)
2. {
3.     float khoangCach;
4.
5.     khoangCach=sqrt( (A.x-B.x)*(A.x-B.x)+
6.                     (A.y-B.y)*(A.y-B.y));
7.     return khoangCach;
8. }
```

## Câu 7 – Vị trí tương đối hình tròn

```
1. void TuongDoi(DUONG_TRON c1, DUONG_TRON c2)
2. {
3.     float d;
4.
5.     d = TinhKhoangCach(c1.O, c2.O);
6.     if((abs(c1.R-c2.R)<d)&&(d<c1.R+c2.R))
7.         cout << "Cat nhau";
8.     if ( (d == abs(c1.R - c2.R)) ||
           (d == c1.R + c2.R))
9.         cout << "Tiep xuc";
10.    if ( (d > c1.R + c2.R) ||
          (d < abs(c1.R - c2.R)))
11.        cout << "khong giao nhau";
12.}
```

## Câu 8 – Rút gọn hỗn số

- Viết chương trình nhập vào một hỗn số, rút gọn hỗn số đó và xuất kết quả.

cuu duong than cong . com

cuu duong than cong . com

## Câu 8 – Rút gọn hỗn số

```
1. #include <iostream>
2. using namespace std;
3.
4. struct HON_SO
5. {
6.     int nguyen;
7.     int tu;
8.     int mau;
9. };
10.
11. void Nhap(HON_SO &hs);
12. void Xuat(HON_SO hs);
13. HON_SO RutGon(HON_SO hs);
```



## Câu 8 – Rút gọn hỗn số

```
1. void main()
2. {
3.     HON_SO hs;
4.
5.     cout << "Nhap hon so" << endl;
6.     Nhap(hs);
7.
8.     hs = RutGon(hs);
9.
10.    cout << "Rut gon = ";
11.    Xuat(hs);
12.}
```

## Câu 8 – Rút gọn hỗn số

```
1. void Xuat(HON_SO hs)
2. {
3.     cout<<hs.nguyen<<
        "("<<hs.tu<<","<<hs.mau<< " )";
4. }
```

cuu duong than cong . com

## Câu 8 – Rút gọn hỗn số

```
1. HON_SO RutGon(HON_SO hs)
2. {
3.     HON_SO kq = hs;
4.     cuu duong than cong . com
5.     kq.nguyen = hs.nguyen + (hs.tu / hs.mau);
6.     kq.tu = hs.tu % hs.mau;
7.     kq.mau = hs.mau;
8.     cuu duong than cong . com
9.     return kq;
10. }
```