

ĐƯỜNG TRÒN - CIRCLE

- Nguyễn Hoàng Yến Như
- Nguyễn Trần Phúc Nghi
- Nguyễn Trần Phúc An
- Nguyễn Đức Anh Phúc
- Trịnh Thị Thanh Trúc
- KS. Hồ Thái Ngọc
- KS. Cao Bá Kiệt
- KS. Lê Ngọc Huy
- CN. Bùi Cao Doanh
- CN. Nguyễn Trọng Thuận
- KS. Phan Vĩnh Long
- KS. Nguyễn Cường Phát
- ThS. Nguyễn Hoàng Ngân
- ThS. Đỗ Văn Tiến
- ThS. Nguyễn Hoàn Mỹ
- ThS. Dương Phi Long
- ThS. Trương Quốc Dũng
- ThS. Nguyễn Thành Hiệp
- ThS. Nguyễn Võ Đăng Khoa
- ThS. Võ Duy Nguyên
- ThS. Trần Việt Thu Phương
- TS. Nguyễn Tấn Trần Minh Khang

Khai báo kiểu dữ liệu

```
101.struct Diem
102.{
103.|    float x;
104.|    float y;
105.};
106.typedef struct Diem DIEM;
```

Khai báo kiểu dữ liệu

```
101.struct DuongTron
102.{
103.|    DIEM I;
104.|    float R;
105.};
106.typedef struct DuongTron DUONGTRON;
```

Nhập đường tròn

— Định nghĩa hàm

```
101. void Nhap(DIEM &P)
102. {
103.     cout << "Nhap x: ";
104.     cin >> P.x;
105.     cout << "Nhap y: ";
106.     cin >> P.y;
107. }
```

Nhập đường tròn

— Định nghĩa hàm

```
101. void Nhap(DUONGTRON &c)
102. {
103.     cout << "Nhap tam: ";
104.     Nhap(c.I);
105.     cout << "Nhap ban kinh:";
106.     cin >> c.R;
107. }
```

Xuất đường tròn

— Định nghĩa hàm

```
101. void Xuat(DIEM P)
102. {
103. |     cout << "\n x=" << P.x;
104. |     cout << "\n y=" << P.y;
105. }
```

Xuất đường tròn

— Định nghĩa hàm

```
101. void Xuat(DUONGTRON c)
102. {
103.     cout << "\n Tam: ";
104.     Xuat(c.I);
105.     cout << "\n Ban kinh:" << c.R;
106. }
```

Tính chu vi

— Định nghĩa hàm

```
101.float ChuVi(DUONGTRON c)
102.{
103.|    return 2 * 3.14 * c.R;
104.}
```

$$P_c = 2\pi R$$

Tính diện tích

— Định nghĩa hàm

```
101.float DienTich(DUONGTRON c)
102.{
103.|    return 3.14 * c.R * c.R;
104.}
```

$$S_c = \pi R^2$$

Kiểm tra một điểm thuộc đường tròn

— Định nghĩa hàm

```
101.float  KhoangCach(DIEM P,DIEM Q)
102.{
103.    |    return sqrt((P.x-Q.x)*(P.x-Q.x)
104.    |                +(P.y-Q.y)*(P.y-Q.y));
105.}
```

Kiểm tra một điểm thuộc đường tròn

— Định nghĩa hàm

```
101.int KtThuoc(DUONGTRON c,DIEM P)
102.{
103.    float kc = KhoangCach(c.I, P);
104.    if(kc<=c.R)
105.        return 1;
106.    return 0;
107.}
```

Vị trí tương đối điểm và đường tròn

- Xét vị trí tương đối giữa một điểm và một đường tròn (nằm trong đường tròn, nằm trên đường tròn, nằm ngoài đường tròn).
- Giá trị trả về: Hàm trả về một trong các giá trị sau:
 - + 0. Điểm nằm trong đường tròn.
 - + 1. Điểm nằm trên đường tròn.
 - + 2. Điểm nằm ngoài đường tròn.
- Khai báo hàm

```
101.int TuongDoi(DUONGTRON,DIEM);
```

Vị trí tương đối của điểm và đường tròn

— Định nghĩa hàm

```
101.int TuongDoi(DUONGTRON c,DIEM P)
102.{
103.    float kc = KhoangCach(c.I,P);
104.    if(kc<c.R)
105.        return 0;
106.    if(kc==c.R)
107.        return 1;
108.    return 2;
109.}
```

- Giá trị trả về: Hàm trả về một trong các giá trị sau:
- + 0. Điểm nằm trong đường tròn.
 - + 1. Điểm nằm trên đường tròn.
 - + 2. Điểm nằm ngoài đường tròn.

Vị trí tương đối đường thẳng và Circle

- Xét vị trí tương đối giữa một đường thẳng và một đường tròn (không cắt, tiếp tuyến, cắt đường tròn).
- Giá trị trả về: Hàm trả về một trong các giá trị sau:
 - + 0. Đường thẳng không cắt đường tròn.
 - + 1. Đường thẳng tiếp xúc với đường tròn.
 - + 2. Đường thẳng cắt đường tròn tại hai điểm.
- Khai báo hàm.

```
101.int TuongDoi(DUONGTRON, DUONGTHANG);
```

Vị trí tương đối đường thẳng và Circle

— Định nghĩa hàm

```
101.float  KhoangCach(DUONGTHANG d,DIEM P)
102.{
103.|    float tu = abs(d.a*P.x + d.b*P.y + d.c);
104.|    float mau = sqrt(d.a*d.a + d.b*d.b);
105.|    return tu/mau;
106.}
```

- Giá trị trả về: Hàm trả về một trong các giá trị sau:
- + 0. Đường thẳng không cắt đường tròn.
 - + 1. Đường thẳng tiếp xúc với đường tròn.
 - + 2. Đường thẳng cắt đường tròn tại hai điểm.

Vị trí tương đối đường thẳng và Circle

— Định nghĩa hàm

```
101.int TuongDoi(DUONGTRON c,DUONGTHANG d)
102.{
103.    float kc = KhoangCach(c.I,d);
104.    if(kc>c.R)
105.        return 0;
106.    if(kc==c.R)
107.        return 1;
108.    return 2;
109.}
```

- Giá trị trả về: Hàm trả về một trong các giá trị sau:
- + 0. Đường thẳng không cắt đường tròn.
 - + 1. Đường thẳng tiếp xúc với đường tròn.
 - + 2. Đường thẳng cắt đường tròn tại hai điểm.

Vị trí tương đối của hai đường tròn

— Các trường hợp:

+ Trùng nhau 

+ Rời nhau 

+ Tiếp xúc ngoài 

+ Cắt nhau 

+ Tiếp xúc trong 

+ Chứa trong nhau 

— Hàm trả về một trong 6 giá trị

+ 0. Trùng nhau 

+ 1. Rời nhau 

+ 2. Tiếp xúc ngoài 

+ 3. Cắt nhau 

+ 4. Tiếp xúc trong 

+ 5. Chứa trong nhau 

Vị trí tương đối của hai đường tròn

— Định nghĩa hàm

```
101.float  KhoangCach(DIEM P,DIEM Q)
102.{
103.    |    return sqrt((P.x-Q.x)*(P.x-Q.x)
104.    |                +(P.y-Q.y)*(P.y-Q.y));
105.}
```

Vị trí tương đối của hai đường tròn

```

101.int TuongDoi(DUONGTRON c1,DUONGTRON c2)
102.{
103.    float kc = KhoangCach(c1.I,c2.I);
104.    if(kc==0 && c1.R==c2.R)
105.        return 0;
106.    if(kc>c1.R+c2.R)
107.        return 1;
108.    if(kc==c1.R+c2.R)
109.        return 2;
110.    if(kc<c1.R+c2.R && kc>abs(c1.R-c2.R))
111.        return 3;
112.    if(kc==abs(c1.R-c2.R))
113.        return 4;
114.    return 5;
115.}
    
```

— Hàm trả về một trong 6 giá trị

+ 0. Trùng nhau



+ 1. Rời nhau



+ 2. Tiếp xúc ngoài



+ 3. Cắt nhau



+ 4. Tiếp xúc trong



+ 5. Chứa trong nhau



Chúc các bạn học tốt
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN TP.HCM

Nhóm UIT-Together
Nguyễn Tấn Trần Minh Khang

Chúc các bạn học tốt
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN TP.HCM

Nhóm UIT-Together
Nguyễn Tấn Trần Minh Khang