

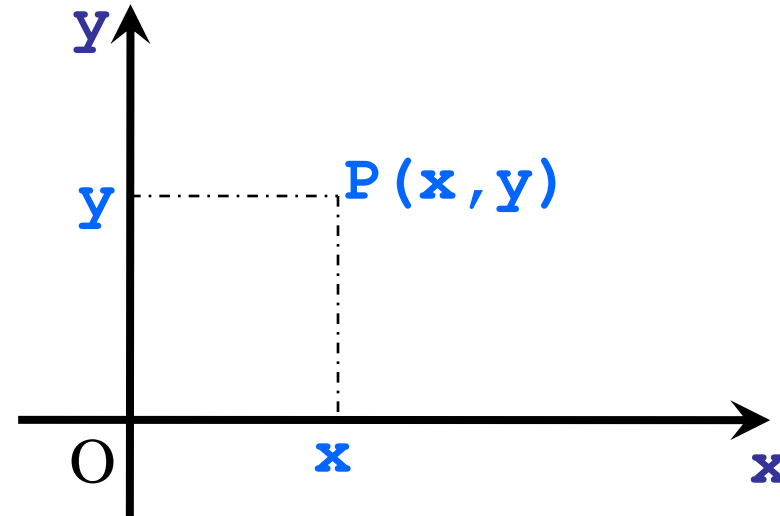
# ĐIỂM

- 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

— Khai báo kiểu dữ liệu

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



# Nhập tọa độ điểm

— 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. }
```

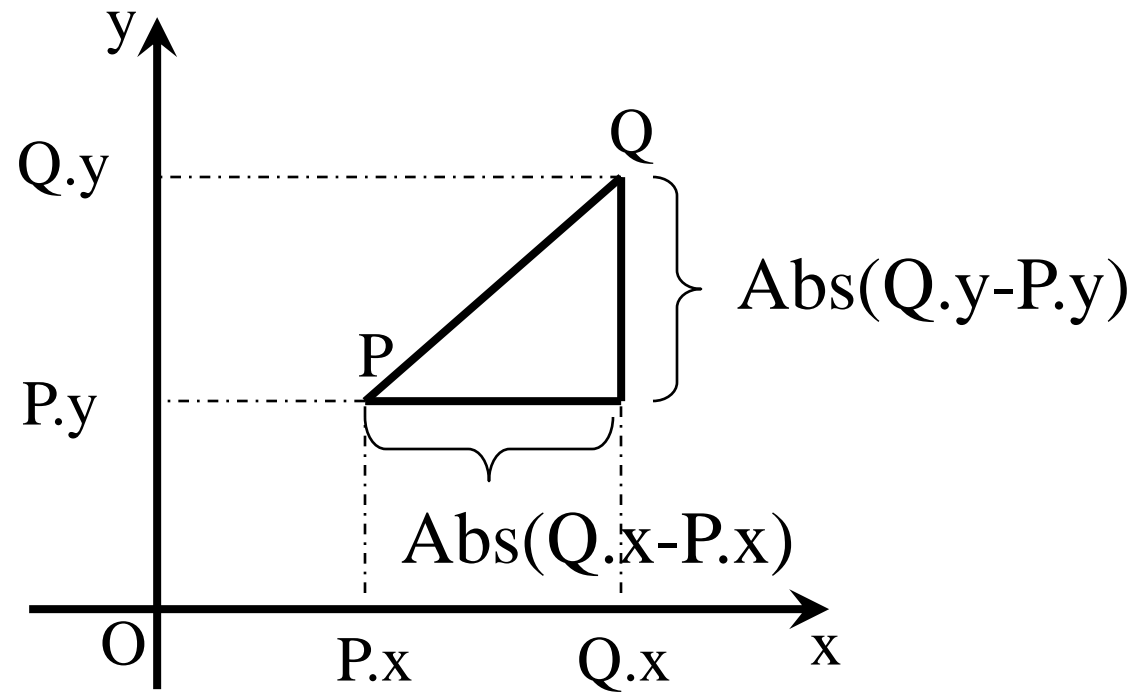
# Xuất tọa độ điểm

## — Định nghĩa hàm

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

```
106. void Xuat(DIEM P)
107. {
108.     cout << "(" << P.x << ", " << P.y << ")";
109. }
```

# Tính khoảng cách



# Tính khoảng cách

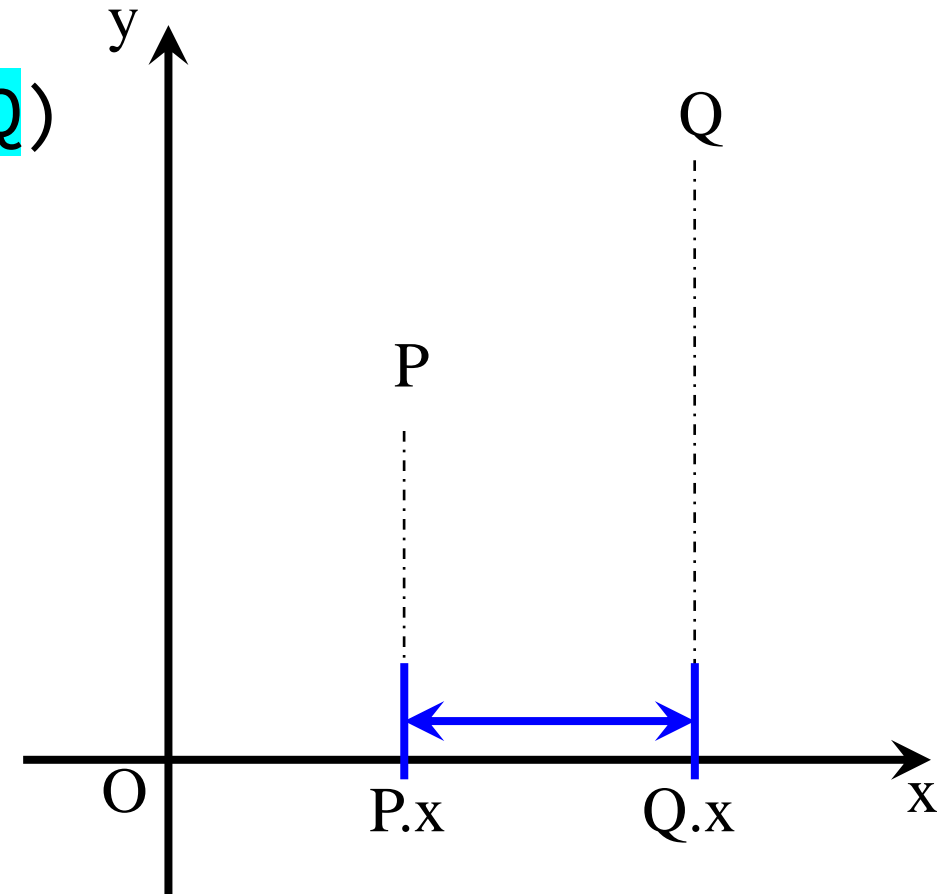
— Định nghĩa hàm

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

# Khoảng cách theo phương Ox

— Định nghĩa hàm

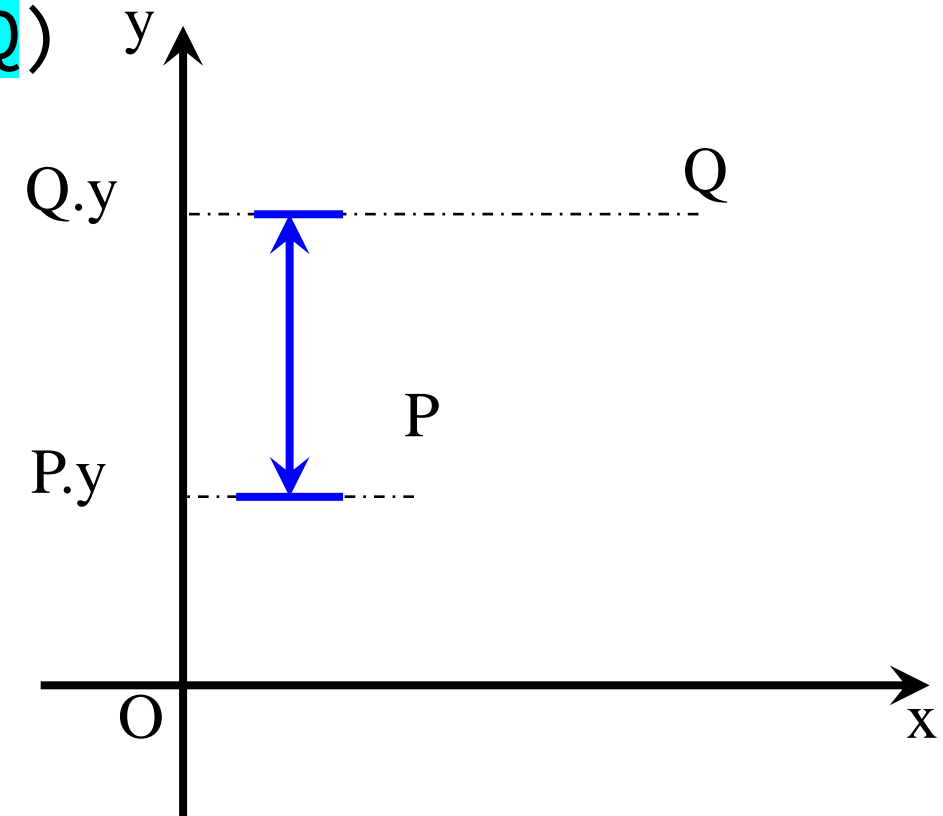
```
101.float  KhoangCachX(DIEM P,DIEM Q)
102.{
103.|      return abs(Q.x-P.x);
104.}
```



# Khoảng cách theo phương Oy

— Định nghĩa hàm

```
101.float  KhoangCachY(DIEM P,DIEM Q)
102.{
103.|    return abs(Q.y-P.y);
104.}
```



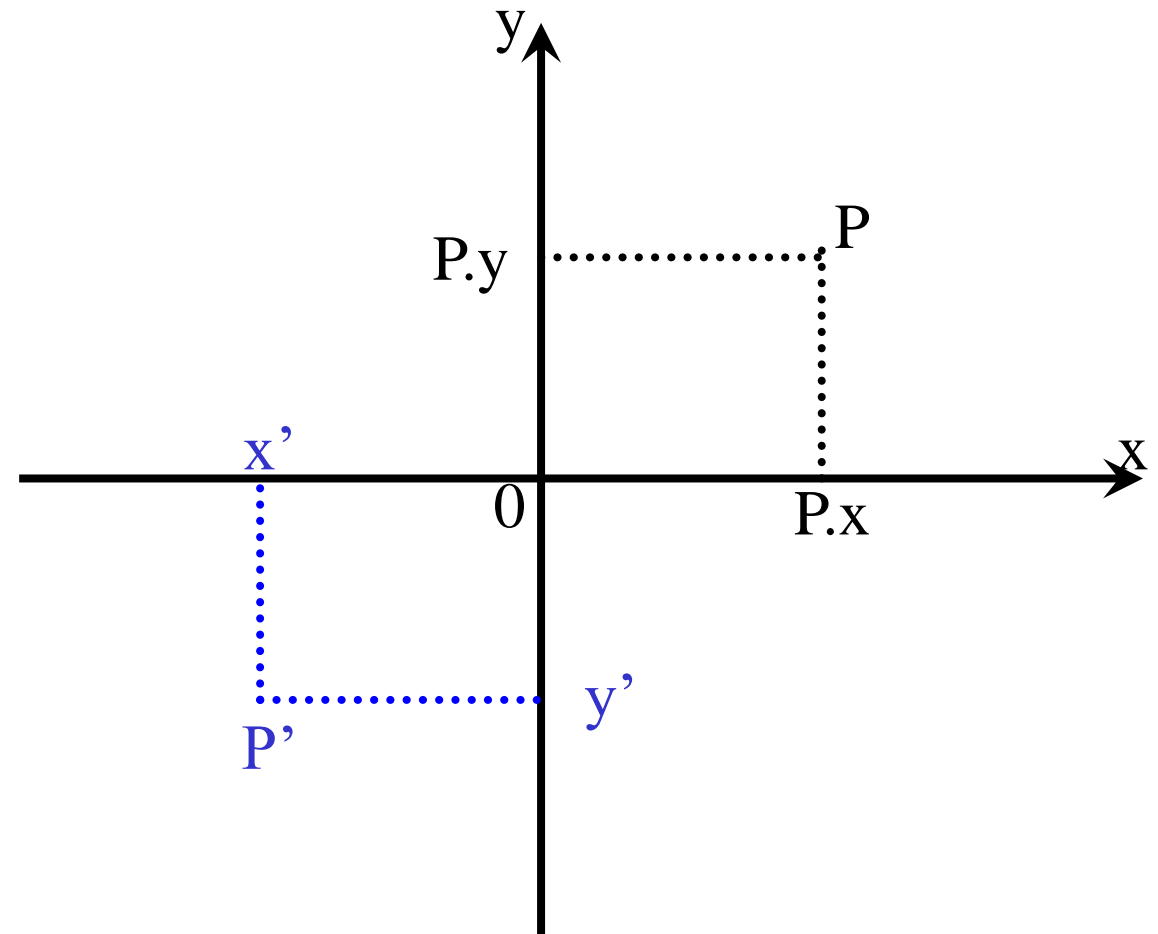


# Điểm đối xứng qua gốc tọa độ

— Định nghĩa hàm

```

101. DIEM DoiXungGoc(DIEM P)
102. {
103.     DIEM temp;
104.     temp.x = -P.x;
105.     temp.y = -P.y;
106.     return temp;
107. }
    
```

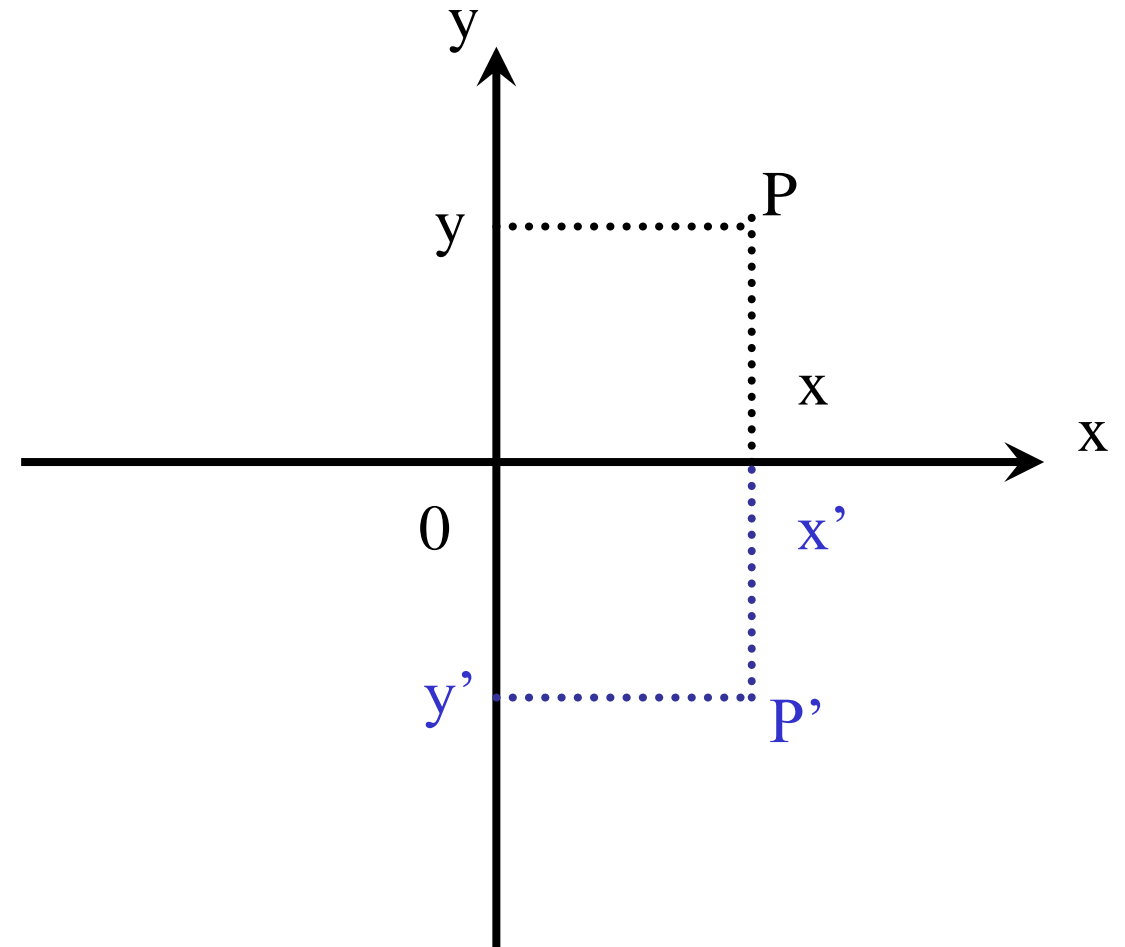


# Điểm đối xứng qua trục hoành

## — Định nghĩa hàm

```

101. DIEM DoiXungHoanh(DIEM P)
102. {
103.     DIEM temp;
104.     temp.x = P.x;
105.     temp.y = -P.y;
106.     return temp;
107. }
    
```

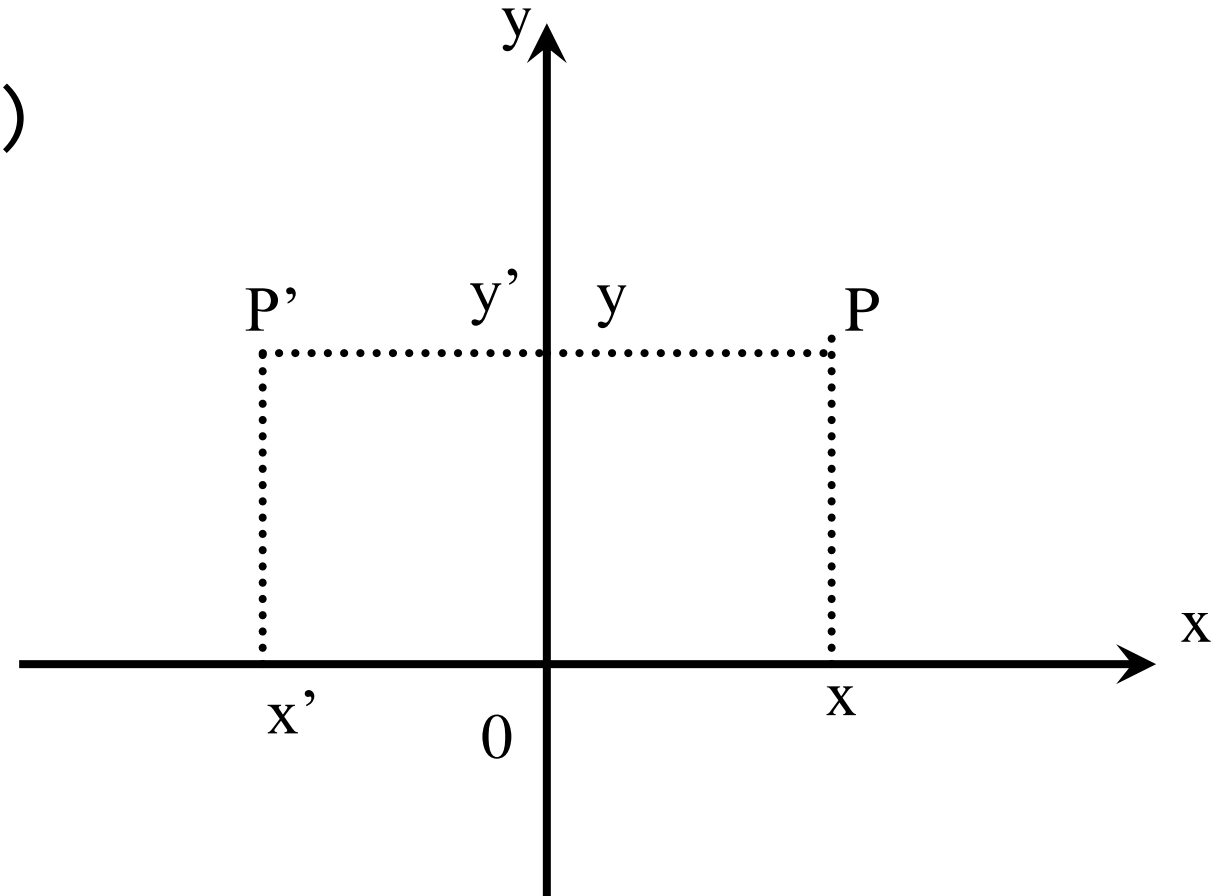


# Điểm đối xứng qua trục tung

## — Định nghĩa hàm

```

101. DIEM DoiXungTung(DIEM P)
102. {
103.     DIEM temp;
104.     temp.x = -P.x;
105.     temp.y = P.y;
106.     return temp;
107. }
    
```

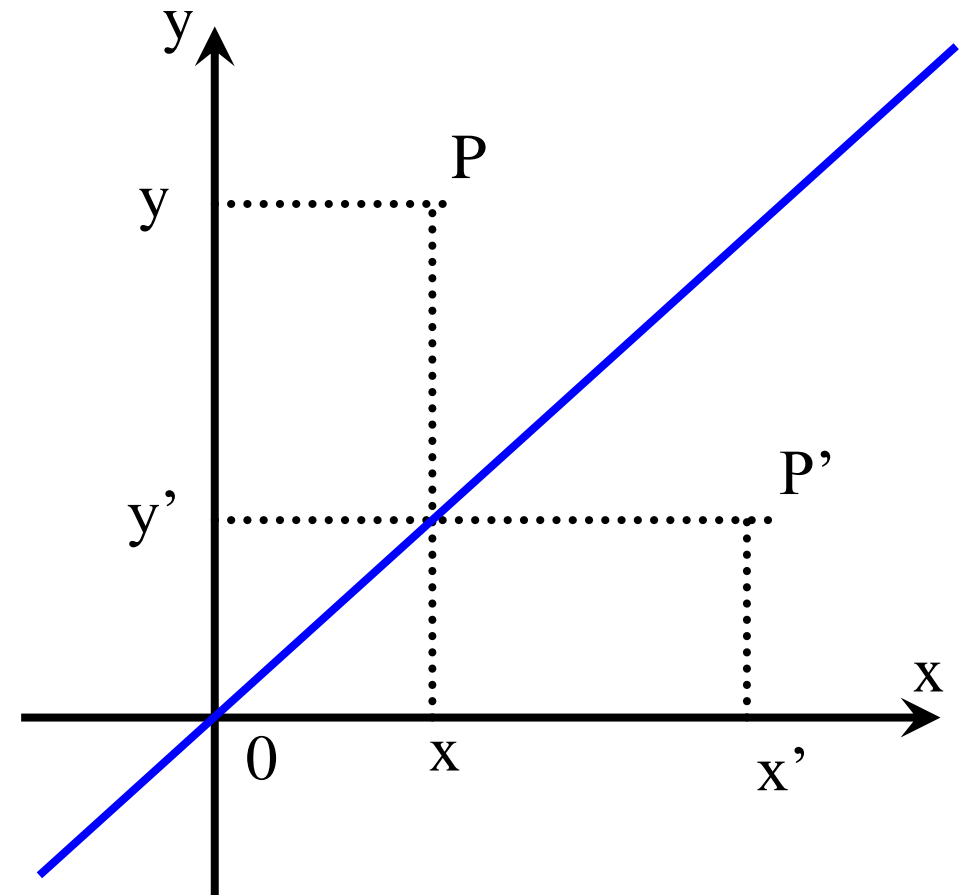


# Đối xứng qua đường phân giác thứ nhất

## — Định nghĩa hàm

```

101. DIEM DoiXungPhanGiac1(DIEM P)
102. {
103.     DIEM temp;
104.     temp.x = P.y;
105.     temp.y = P.x;
106.     return temp;
107. }
    
```

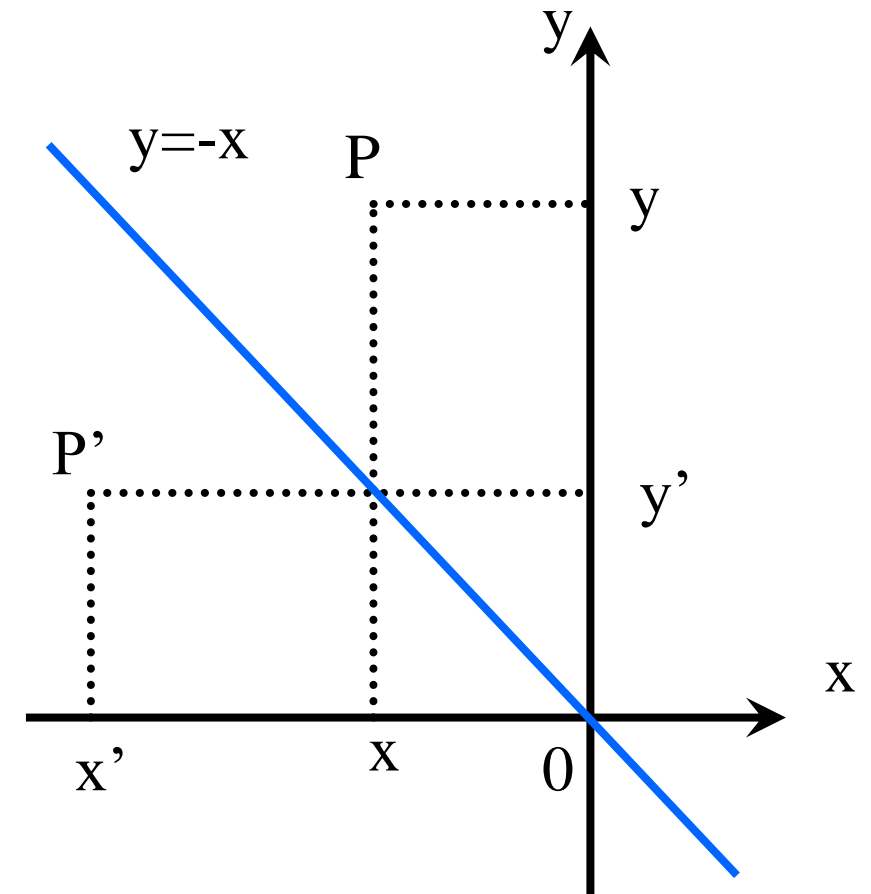


# Đối xứng qua đường phân giác thứ hai

— Định nghĩa hàm

```

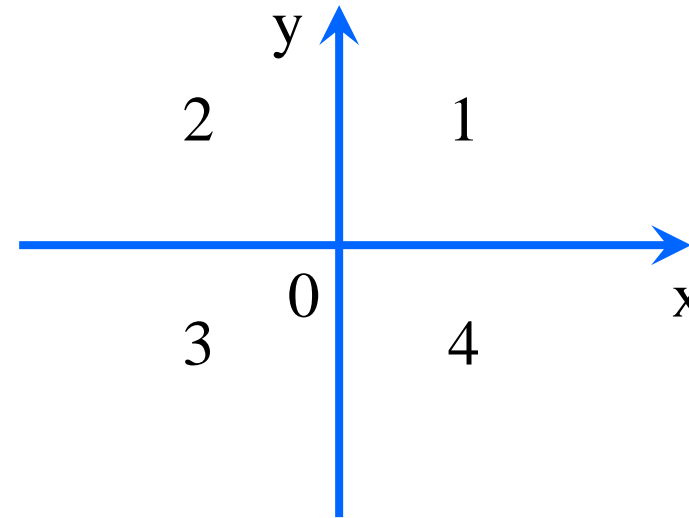
101. DIEM DoiXungPhanGiac2(DIEM P)
102. {
103.     DIEM temp;
104.     temp.x = -P.y;
105.     temp.y = -P.x;
106.     return temp;
107. }
    
```



# Kiểm tra thuộc phần tư thứ 1

— Định nghĩa hàm

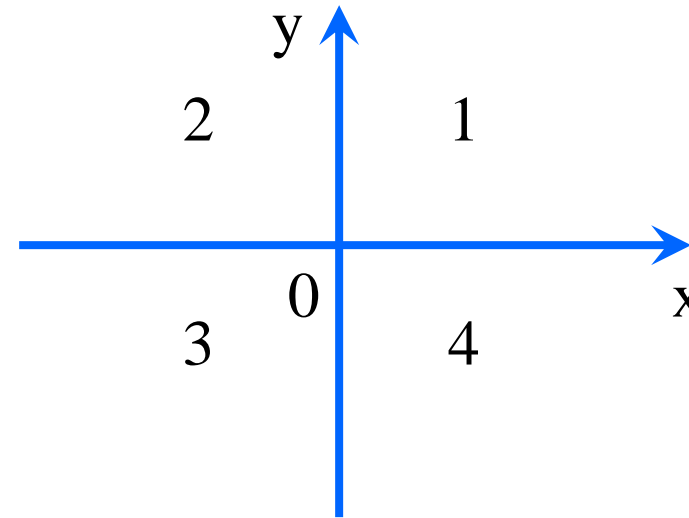
```
101.int ktThuoc1(DIEM P)
102.{
103.    if(P.x>0 && P.y>0)
104.        return 1;
105.    return 0;
106.}
```



# Kiểm tra thuộc phần tư thứ 2

— Định nghĩa hàm

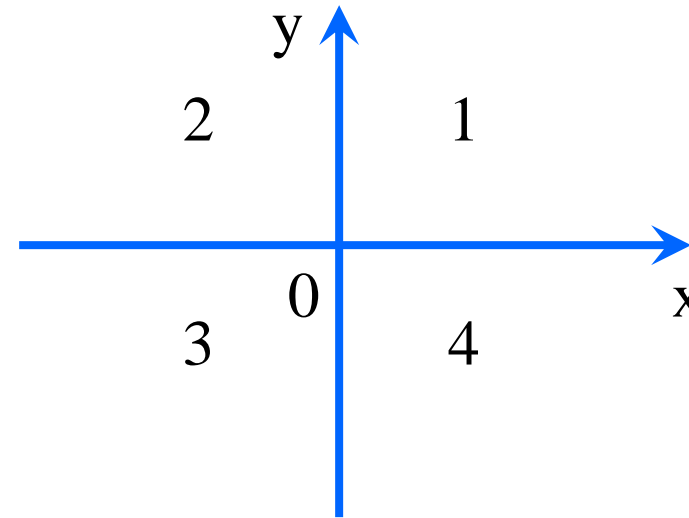
```
101.int ktThuoc2(DIEM P)
102.{
103.    if(P.x<0 && P.y>0)
104.        return 1;
105.    return 0;
106.}
```



# Kiểm tra thuộc phần tư thứ 3

— Định nghĩa hàm

```
101.int ktThuoc3(DIEM P)
102.{
103.    if(P.x<0 && P.y<0)
104.        return 1;
105.    return 0;
106.}
```

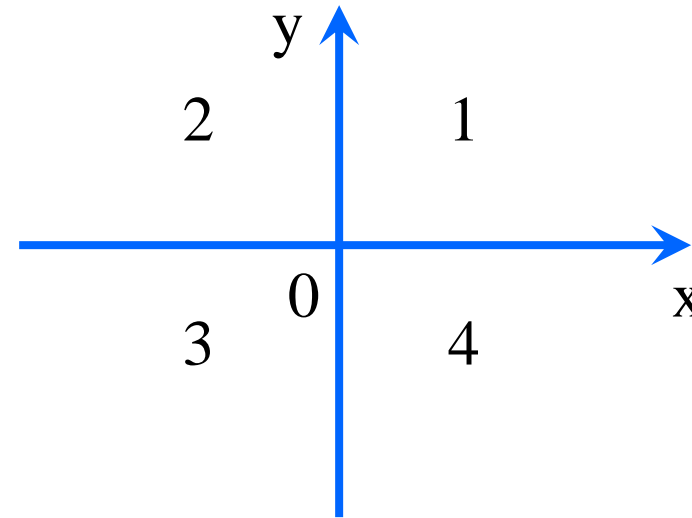




# Kiểm tra thuộc phần tư thứ 4

— Định nghĩa hàm

```
101.int ktThuoc4(DIEM P)
102.{
103.    if(P.x>0 && P.y<0)
104.        return 1;
105.    return 0;
106.}
```



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

**Nhóm UIT-Together**  
**Nguyễn Tấn Trần Minh Khang**