

Mục tiêu:

1. Tổ chức dữ liệu kiểu cấu trúc và danh sách đặc gồm các phần tử kiểu cấu trúc.
2. Thao tác trên kiểu Cấu trúc
3. Truyền tham biến và tham trị kiểu Cấu trúc cho chương trình con
4. Cài đặt một số ứng dụng trên kiểu cấu trúc.

Vấn đề 1: Nhận từ bàn phím phân số $\frac{a}{b}$ và hiển thị lên màn hình phân số tối giản.

<i>input</i>	<i>output</i>
$\frac{2}{4}$	$\frac{1}{2}$
$\frac{1}{7}$	$\frac{1}{7}$
$\frac{-3}{6}$	$\frac{-1}{2}$

Hướng dẫn:

1. Định nghĩa PS là kiểu phân số gồm 2 thành phần **tử số** (là một số nguyên) và **mẫu số** (là một số tự nhiên khác 0).
2. Cài đặt hàm **void input(PS &phanso)** nhận vào một phân số **phanso**;
3. Cài đặt hàm **void output(PS phanso)** hiển thị phân số **phanso** lên màn hình;
4. Cài đặt hàm **int ucln(int a, int b)** trả về ước số chung lớn nhất của 2 số nguyên không âm a và b;
5. Cài đặt hàm **void toigian(PS &phanso)** để tối giản phân số **phanso**;
6. Tham khảo chương trình bên dưới để thực hiện kiểm tra tính đúng của các hàm vừa cài đặt ở trên.

```
#include <stdio.h>
typedef struct ps
{
    // định nghĩa phân số
} PS;

void input(PS &phanso);
void output(PS phanso);
int ucln(int a, int b);
void toigian(PS &phanso);
int main()
{
    PS p;
    printf("\nNhap phan so:");    input(p);
    printf("\nPhan so vua nhap:");    output(p);
    toigian(p);
    printf("\nPhan so sau khi toi gian:");    output(p);
    return 0;
}
```

Vấn đề 2: Đa giác lồi n đỉnh có tọa độ các đỉnh được lưu trong mảng A. Tính và hiển thị lên màn hình diện tích của đa giác lồi đó.

Hướng dẫn:

1. Định nghĩa kiểu cấu trúc mô tả một điểm (Point) có tọa độ nguyên trong hệ tọa độ đề-các Oxy. Cấu trúc **Point** được định nghĩa như sau:

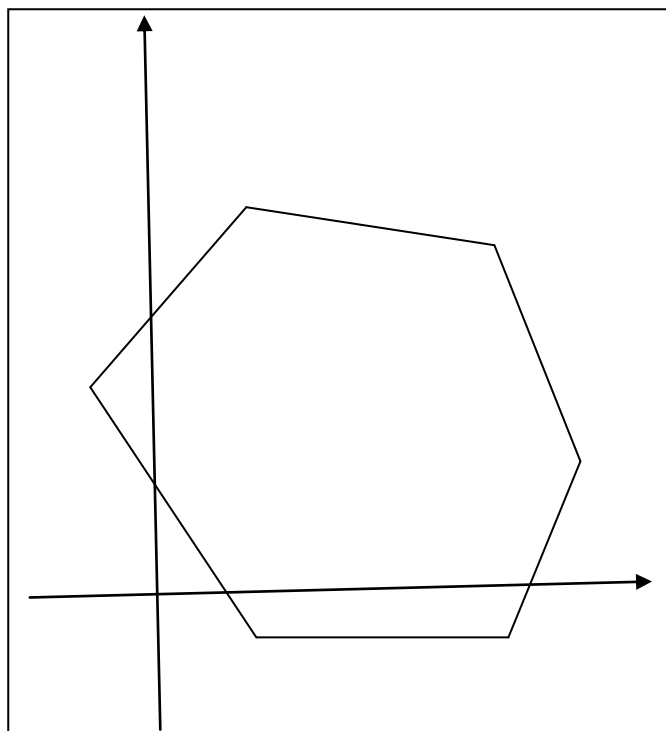
```
typedef struct point
{
    int x;
    int y;
}Point;
```

2. Viết hàm trả về độ dài đoạn thẳng P_1P_2 khi biết tọa độ 2 điểm đầu mút.

float P_1P_2 (Point P_1 , Point P_2);

Ghi chú: Công thức tính độ dài đoạn thẳng AB:

$$\overline{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$



3. Viết hàm trả về diện tích của tam giác $P_1P_2P_3$ khi biết tọa độ các đỉnh của chúng hoặc trả về 0 khi P_1, P_2, P_3 thẳng hàng.

float $SP_1P_2P_3$ (Point P_1 , Point P_2 , Point P_3);

4. Tổ chức danh sách đặc (mảng) A chứa tọa độ n đỉnh của đa giác lồi. Thực hiện tính diện tích các tam giác $A_0A_1A_2, A_0A_2A_3, A_0A_3A_4, \dots, A_{n-2}A_{n-1}A_n, A_{n-1}A_nA_0$. Khi đó diện tích đa giác lồi bằng tổng diện tích của n-2 tam giác đó.

- Định nghĩa đa giác

```
#define max 1000
```

```
typedef Point PointArray[max];
```

- Viết hàm trả về diện tích đa giác

float S (PointArray A, int n);

Báo cáo:

Lập báo cáo Input và output cho 3 trường hợp n=3, 6, 10

n	A	S
3	x y	
6	x y	
10	x y	

Mở rộng:

5. Thực hiện kiểm tra dữ liệu vào để đảm bảo đa giác nhận được là đa giác lồi.
6. Trong trường hợp đa giác không lồi. Hãy chia đa giác đã cho tập các đa giác lồi.