|  |
| --- |
| **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 và hiển thị lên màn hình phân số tối giản.

|  |  |
| --- | --- |
| ***input*** | ***output*** |
|  |  |
|  |  |
|  |  |

***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).

typedef struct PhanSo{

int tuso, mauso;

}ps;

1. Cài đặt hàm ***void input(PS &phanso)*** nhận vào một phân số ***phanso***;

void input(PhanSo &ps) {

printf("Nhap tu so: ");

scanf("%d", &ps.tuso);

printf("Nhap mau so: ");

scanf("%d", &ps.mauso);

}

1. Cài đặt hàm ***void output(PS phanso)*** hiển thị phân số ***phanso*** lên màn hình;

void output(PhanSo ps) {

printf("%d/%d", ps.tuso, ps.mauso);

}

1. Cài đặt hàm ***int ucln(int a, int b)*** trả về ước số chung lớn nhất cùa 2 số nguyên không âm a và b;

int ucln(int a, int b) {

if (b == 0)

return a;

else

return ucln(b, a % b);

}

1. Cài đặt hàm ***void toigian(PS &phanso)*** để tối giản phân số ***phanso***;

void toigian(PhanSo &ps) {

int gcd = ucln(ps.tuso, ps.mauso);

ps.tuso /= gcd;

ps.mauso /= gcd;

}

1. 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;**

1. Viết hàm trả về độ dài đoạn thẳng P1P2 khi biết tọa độ 2 điểm đầu mút.

***float P1P2(Point P1, Point P2);***

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

=

float P1P2(Point p1, Point p2) {

float dx = 1.0 \* p1.x - p2.x;

float dy = 1.0 \* p1.y - p2.y;

return sqrt(dx \* dx + dy \* dy);

}

1. Viết hàm trả về diện tích của tam giác P1P2P3 khi biết tọa độ các đỉnh của chúng hoặc trả về 0 khi P1,P2, P3 thẳng hàng.

***float SP1P2P3(Point P1, Point P2, Point P3);***

float SP1P2P3(Point p1, Point p2, Point p3) {

int dientich = 0;

float a = P1P2(p1,p2);

float b = P1P2(p2,p3);

float c = P1P2(p1,p3);

if (a + b <= c || a + c <= b || b + c <= a) {

return 0;

}

float p = (a + b + c) / 2;

dientich = sqrt(p \* (p - a) \* (p -b) \* (p - c));

return dientich;

}

1. Tổ chức danh sách đặc (mảng) A chứa tọa độ n đỉnh của đa giác lồi. Thực hiện tính diện tích các tam giác A0A1A2, A0A2A3,A0A3A4, ..., An-2An-1An, An-1AnA0. 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);***

float S(PointArray A, int n) {

float S = 0;

for (int i = 1; i < n -1; i++) {

S += SP1P2P3(A[0], A[i], A[i+1]);

}

return S;

}

***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:***

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