

## Question 1

Not answered

Mark 0.00 out of 10.00

### [Struct]

Struct cũng là cấu trúc cho phép định nghĩa các kiểu dữ liệu có cấu trúc: ngoài dữ liệu còn kèm theo cả các hàm xử lý dữ liệu đó. Ví dụ:

- [Hình chữ nhật](#) có dữ liệu là chiều dài, chiều rộng và các hàm như tính chu vi, diện tích, ...
- Vector trong hệ tọa độ Đề-Các có dữ liệu là cặp tọa độ  $x$  và  $y$ ; các hàm tính tổng, tích vô hướng,... của một vector với một vector khác.

### Khai báo cấu trúc

Ta có khai báo một cấu trúc mới như sau,

```
struct <tên cấu trúc> {<br>    <kiểu dữ liệu 1> biến 1;<br>    <kiểu dữ liệu 2> biến 2;<br>    ...<br>    <hàm 1><br>    <hàm 2><br>    ...<br>};
```

Ví dụ, ta có thể khai báo cấu trúc **Vector** như sau,

```
struct Vector {<br>    double x;<br>    double y;<br>    void printVector {<br>        // in ra giá trị của một Vector<br>        cout << x << " " << y << endl;<br>    }<br>};
```

### Sử dụng cấu trúc

Ta có thể truy cập vào các trường dữ liệu hoặc thực thi các hàm của một cấu trúc như sau.

Ví dụ, với cấu trúc **Vector** đã được khai báo ở trên,

```
Vector v;    // Khai báo biến v kiểu Vector<br>v.x = 1;    // Gán giá trị cho trường x của biến v<br>v.y = 2;    // Gán giá trị cho trường y của biến v<br>v.printVector(); // In ra giá trị của Vector v
```

### Hàm khởi tạo

Hàm khởi tạo là hàm thành viên đặc biệt có nhiệm vụ khởi tạo các trường dữ liệu trong một cấu trúc. Hàm khởi tạo thường có các tính chất sau,

- Được gọi tự động khi khai báo hoặc cấp phát biến động;
- Trùng tên với tên cấu trúc;
- Không có kiểu trả về.

Ví dụ,

```
struct Vector {<br>    double x;<br>    double y;<br>    Vector (double _x, double _y){<br>        x = _x;<br>        y = _y;<br>    }<br>    void printVector {<br>        // in ra giá trị của một Vector<br>        cout << x << " " << y << endl;<br>    }<br>};
```

Như vậy, ta có thể khởi tạo giá trị của các trường dữ liệu một cách đơn giản như sau,

```
Vector v1(1.3,2.7);<br>Vector v2(2.3,4.1);
```

### Bài tập

Viết chương trình tính tổng 2 vector trong hệ tọa độ Đề-Các (sử dụng cấu trúc struct như ở trên).

### Đầu vào

Đầu vào từ bàn phím gồm 2 dòng. Mỗi dòng chứa 2 số thực biểu diễn tọa độ  $x$  và  $y$  của một vector. Các số nguyên trên cùng một dòng cách nhau bởi một dấu cách và có giá trị tuyệt đối không vượt quá 100.

### Đầu ra

In ra màn hình 2 số thực cách nhau bởi một dấu cách, biểu diễn tọa độ  $x$  và  $y$  của vector tổng của 2 vừa nhập.

For example:

| Input      | Result |
|------------|--------|
| 2 4<br>3 5 | 5 9    |

Answer:

Reset answer

```
1 #include <iostream>
2 using namespace std;
3
4 struct Vector {
5     double x;
6     double y;
7
8     //Hàm tạo - Tạo ra một Vector với thành phần dữ liệu đã biết trước
9     Vector (double _x, double _y){
10         x = _x;
11         y = _y;
12     }
13
14     //Hàm in ra màn hình một Vector
15     void printVector(){
16         cout << x << " " << y << endl;
17     }
18 };
19
20 //Hàm tính tổng 2 Vector trả về Vector tổng sau khi tính toán
21 Vector addVector(const Vector &v1, const Vector &v2)
22 {
23     Vector sum;
24     // Them code cua ban o day
25     return sum;
26 }
27
28 int main()
29 {
30     // Them code cua ban o day
31 }
```

|   | Input               | Expected | Got      |   |
|---|---------------------|----------|----------|---|
| ✓ | 2 4<br>3 5          | 5 9      | 5 9      | ✓ |
| ✓ | 3.2 5.3<br>10.3 2.4 | 13.5 7.7 | 13.5 7.7 | ✓ |

Passed all tests! ✓

Back to Course