

Question 1

Incorrect

Mark 0.00 out of 10.00

Struct

Struct là [cấu trúc](#) cho phép định nghĩa các kiểu dữ liệu có [cấu trúc](#): dữ liệu kèm theo 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 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	5 9
3 5	

Answer: (penalty regime: 0 %)

Reset answer

```

1  #include <iostream>
2  using namespace std;
3
4  struct Vector {
5      double x;
6      double y;
7      Vector (double _x, double _y){
8          x = _x;
9          y = _y;
10     }
11     void printVector {
12         cout << x << " " << y << endl;
13     }
14 };
15
16 Vector addVector(Vector v1, Vector v2)
17 {
18     // Them code cua ban o day
19 }
20
21 int main()
22 {
23     // Them code cua ban o day
24 }
```

Syntax Error(s)

__tester__.cpp:11:10: error: variable or field 'printVector' declared void

```

11 |     void printVector {
    |           ^~~~~~
```

__tester__.cpp:13:5: error: expected ';' at end of member declaration

```

13 |     }
    |     ^
    |     ;
```

__tester__.cpp: In function 'Vector addVector(Vector, Vector)':

__tester__.cpp:19:1: error: no return statement in function returning non-void [-Werror=return-type]

```

19 | }
    | ^
```

cc1plus: all warnings being treated as errors

Incorrect

Marks for this submission: 0.00/10.00.

Question 2

Not answered

Mark 0.00 out of 10.00

[Rectangle]

Khai báo [cấu trúc](#) dữ liệu/ lớp [hình chữ nhật](#) gồm độ dài 2 cạnh (kiểu số nguyên).

Viết chương trình nhập vào 2 số nguyên là chiều dài và chiều rộng của một [hình chữ nhật](#) và tính chu vi và diện tích của [hình chữ nhật](#) đó

Đầu vào

Một dòng duy nhất từ bàn phím chứa 2 số nguyên dương có giá trị không vượt quá 100, cách nhau bởi một dấu cách, là chiều dài và chiều rộng của một [hình chữ nhật](#).

Đầu ra

In ra màn hình một dòng duy nhất chứa 2 số nguyên là chu vi và diện tích của [hình chữ nhật](#) đó.

For example:

Input	Result
3 2	10 6

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2 using namespace std;
3
4 struct rectangle{
5     int length, high;
6 };
7
8 ///Ham tinh Chu vi của một hình chữ nhật, hàm không làm thay đổi HCN ban đầu
9 int getPerimeter(const rectangle &rec){
10     // them code của bạn ở đây
11 }
12
13 ///Ham tinh diện tích của một hình chữ nhật, hàm không làm thay đổi HCN ban đầu
14 int getArea(const rectangle &rec){
15     // them code của bạn ở đây
16 }
17
18 int main() {
19     // them code của bạn ở đây
20 }
```

Question 3

Not answered

Mark 0.00 out of 10.00

[Cylinder]

Khai báo [cấu trúc](#) (lớp) dữ liệu [hình trụ](#) gồm: bán kính và chiều cao (kiểu số thực).

Hãy viết chương trình trong đó có các hàm tính diện tích bề mặt (diện tích toàn phần) và thể tích của [hình trụ](#) với đầu vào là một [cấu trúc](#) dữ liệu [hình trụ](#).

Đầu vào

Đầu vào từ bàn phím gồm một dòng duy nhất chứa 2 số nguyên dương có giá trị không vượt quá 100, cách nhau bởi một dấu cách, là bán kính và chiều cao [hình trụ](#).

Đầu ra

In ra màn hình diện tích bề mặt và thể tích theo thứ tự (sử dụng lại những hàm vừa viết, $\pi = 3.14$).

(Kết quả làm tròn đến 2 chữ số sau dấu phẩy).

For example:

Input	Result
1 2	18.84 6.28

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2 using namespace std;
3
4 struct cylinder{
5     int radius,high;
6 };
7
8 //Hàm tính diện tích bề mặt,
9 //hàm này không làm thay đổi thông số của hình trụ ban đầu
10 double getSurfaceArea(const cylinder &c){
11     // them code của bạn ở đây
12 }
13
14 //Hàm tính thể tích hình trụ
15 //hàm này không làm thay đổi thông số của hình trụ ban đầu
16 double getVolume(const cylinder &c){
17     // them code của bạn ở đây
18 }
19
20 int main() {
21     // them code của bạn ở đây
22 }
23
```

Question 4

Not answered

Mark 0.00 out of 10.00

[Date]

Khai báo kiểu dữ liệu [cấu trúc](#) (lớp) biểu diễn [ngày tháng](#) năm.

Nhập vào [ngày tháng](#) năm, hãy viết hàm (phương thức) tính và hiển thị ngày kế tiếp của [ngày tháng](#) năm vừa nhập ([ngày tháng](#) năm biểu diễn dưới dạng *dd/mm/yyyy*).

Nếu [ngày tháng](#) năm không tồn tại, in ra dòng chữ *INVALID*.

Đầu vào

Đầu vào từ bàn phím chứa dữ liệu ngày, tháng, năm dưới dạng *dd/mm/yyyy*.

Đầu ra

In ra màn hình ngày kế tiếp của ngày vừa nhập dưới dạng *dd/mm/yyyy*. Nếu [ngày tháng](#) không tồn tại, in ra *INVALID*.

For example:

Input	Result
14/11/2016	15/11/2016

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2 using namespace std;
3
4 struct date{
5     int day, month, year;
6 };
7
8 ///Hàm tính và trả về ngày kế tiếp của ngày hiện tại
9 ///Hàm không làm thay đổi giá trị của ngày hiện tại
10 date getNextDay(const date &currentDate){
11     // them code của bạn ở đây
12 }
13
14 int main() {
15     // them code của bạn ở đây
16 }
17
```

Question 5

Not answered

Mark 0.00 out of 10.00

[Fraction]

Khai báo kiểu dữ liệu [cấu trúc](#) (lớp) biểu diễn phân số với tử số và mẫu số nguyên.

Nhập vào giá trị của 2 phân số, viết chương trình tính tổng 2 phân số ở dạng tối giản.

Đầu vào

Đầu vào từ bàn phím gồm 4 số nguyên cách nhau bởi một dấu cách, tương ứng với tử số và mẫu số của 2 phân số. Các số nguyên từ đầu vào có giá trị tuyệt đối không vượt quá 100.

Đầu ra

In ra màn hình tổng của 2 phân số in ra dưới dạng phân số tối giản a/b .

For example:

Input	Result
1 2 3 4	5/4

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2 using namespace std;
3
4 struct fraction{
5     int numerator, denominator;
6     fraction (int _numerator, int _denominator){
7         // them code cua ban o day
8     }
9 };
10
11 fraction addconst (const fraction &a, const fraction &b){
12     // them code cua ban o day
13 }
14
15 int main() {
16     // them code cua ban o day
17 }
18
```

Question 6

Not answered

Mark 0.00 out of 10.00

[Time]

Khai báo cấu trúc thời gian *Time* bao gồm 3 biến *hour*, *minute*, *second* tương ứng với **giờ**, **phút**, **giây** .

Viết hàm khởi tạo `Time(int initHour, int initMinute, int initSecond)` để khởi tạo giá trị giờ, phút, giây cho mốc thời gian và hàm `void getTimeDifference(Time other)` in ra chênh lệch giữa hai mốc thời gian.

Hoàn thiện chương trình dưới đây để nhập vào 2 mốc thời gian và in ra chênh lệch giữa 2 mốc thời gian này.

Đầu vào

Đầu vào từ bàn phím gồm 2 dòng, mỗi dòng gồm 3 số nguyên cách nhau bởi một dấu cách, tương ứng với 2 mốc thời gian.

Đầu ra

In ra màn hình Khoảng cách giữa 2 mốc thời gian in ra dưới dạng *hh : mm : ss*.

For example:

Input	Result
2 5 30	05:10:10
7 15 40	

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2
3 using namespace std;
4
5 struct Time
6 {
7     int hour, minute, second;
8     Time(int initHour, int initMinute, int initSecond)
9     {
10         //your code here
11     }
12     void getTimeDifference(Time other)
13     {
14         //your code here
15     }
16 };
17
18 int main()
19 {
20     int hour, minute, second;
21     cin >> hour >> minute >> second;
22     Time time1(hour,minute,second);
23     cin >> hour >> minute >> second;
24     Time time2(hour,minute,second);
25     time1.getTimeDifference(time2);
26
27     return 0;
28 }
29
```

Back to Course

