

Question 1

Incorrect

Mark 0.00 out of 10.00

[Complex]

Khai báo kiểu dữ liệu biểu diễn số phức *ComplexNumber* bao gồm 2 biến *realPart* và *imaginaryPart* kiểu *double* biểu diễn phần thực và phần ảo của một số phức.

Viết hàm khởi tạo `ComplexNumber(double initRealPart, double initImaginaryPart)` để khởi tạo các giá trị phần thực và phần ảo của một số phức và hàm `ComplexNumber multiply(ComplexNumber other)` trả về tích của số phức hiện tại và một số phức khác.

Đầu vào

Đầu vào từ bàn phím gồm hai dòng, mỗi dòng gồm hai số thực cách nhau bởi một dấu cách biểu diễn phần thực và phần ảo của một số phức. Các số trong đầu vào có giá trị tuyệt đối không vượt quá 100.

Đầu ra

In ra màn hình hai số biểu diễn phần thực và phần ảo của tích hai số phức.

For example:

Input	Result
1 2	-5 10
3 4	

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2
3 using namespace std;
4
5 struct complexNumber
6 {
7     double realPart,imaginaryPart;
8     complexNumber()
9     {
10         realPart = 0;
11         imaginaryPart = 0;
12     }
13     complexNumber(double initRealPart,double initImaginaryPart)
14     {
15         // your code here
16     }
17     complexNumber multiply(complexNumber other)
18     {
19         // your code here
20     }
21 };
22
23 int main()
24 {
25     double realPart,imaginaryPart;
26     cin >> realPart >> imaginaryPart;
27     complexNumber complex1(realPart,imaginaryPart);
28     cin >> realPart >> imaginaryPart;
29     complexNumber complex2(realPart,imaginaryPart);
30     complexNumber product = complex1.multiply(complex2);
31     cout << product.realPart << " " << product.imaginaryPart << endl;
32     return 0;
33 }
34
```

Syntax Error(s)

```
__tester__.cpp: In member function 'complexNumber complexNumber::multiply(complexNumber)':  
__tester__.cpp:20:5: error: no return statement in function returning non-void [-Werror=return-type]  
    20 |     }  
       |     ^
```

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

[Triangle]

Khai báo [cấu trúc](#) dữ liệu tam giác *Triangle* chứa 3 biến *edge1*, *edge2*, *edge3* kiểu *int* là độ dài 3 cạnh của tam giác đó.

Viết hàm khởi tạo `Triangle(int initEdge1, int initEdge2, int initEdge3)` để khởi tạo độ dài các cạnh của tam giác và hàm `double getArea()` trả về diện tích của tam giác đó (nếu tam giác không hợp lệ, trả về -1).

Hoàn thiện chương trình dưới đây nhập vào 3 cạnh của một tam giác và in ra diện tích của tam giác đó.

Đầu vào

Một dòng duy nhất từ bàn phím chứa 3 số nguyên là độ dài 3 cạnh của tam giác

Đầu ra

In ra màn hình diện tích tam giác (làm tròn đến 2 chữ số thập phân sau dấu phẩy). In ra *invalid* nếu 3 số nhập vào không phải là 3 cạnh của một tam giác

For example:

Input	Result
3 4 5	6.00

Answer: (penalty regime: 0 %)

Reset answer

```
1 #include <iostream>
2 #include <iomanip>
3 #include <cmath>
4
5 using namespace std;
6
7 struct triangle
8 {
9     int edge1, edge2, edge3;
10    triangle(int initEdge1, int initEdge2, int initEdge3)
11    {
12        // your code here
13    }
14
15    double getArea()
16    {
17        // your code here
18    }
19 };
20
21 int main()
22 {
23     int edge1, edge2, edge3;
24     cin >> edge1 >> edge2 >> edge3;
25     triangle tri(edge1, edge2, edge3);
26     double area = tri.getArea();
27     if (area == -1) cout << "invalid" << endl;
28     else cout << fixed << setprecision(2) << area << endl;
29     return 0;
30 }
31
```

Question 3

Not answered

Mark 0.00 out of 10.00

[Fraction 2]

Sinh viên thực hiện các yêu cầu sau:

- Định nghĩa **cấu trúc** phân số **Fraction** gồm có:
 - Tử số, mẫu số tương ứng là **a, b** kiểu số nguyên (không âm).
 - Hàm khởi tạo mặc định (gán hai phần bằng 0), và hàm khởi tạo nhận vào tử số và mẫu số.
 - Hàm tính giá trị **double value()** trả về giá trị của phân số tương ứng nếu $b \neq 0$, ngược lại, trả về -1.
 - Hàm **void print()** in ra phân số dưới dạng a/b sau khi đã tối giản phân số. In thêm dấu xuống dòng ở cuối.
- Định nghĩa hàm **Fraction add(Fraction x, Fraction y)** trả về tổng của hai phân số không cần tối giản.

Gợi ý: Cộng hai phân số dùng đúng công thức. Tìm ƯCLN của tử số và mẫu số để tối giản phân số.

Lưu ý: Sinh viên chỉ thực hiện yêu cầu trên mà không viết hàm **main()**.

For example:

Test	Input	Result
Fraction x; x.a = inputA; x.b = inputB; x.print(); cout << fixed << setprecision(2) << x.value();	10 5	2/1 2.00
Fraction x(xa, xb), y(ya, yb); x.print(); y.print(); add(x, y).print();	1 2 2 4	1/2 1/2 1/1

Answer: (penalty regime: 0 %)

Reset answer

```
1 struct Fraction {
2     // your code goes here
3     // Cac bien thanh vien
4
5     // your code goes here
6     // Hai ham khoi tao
7
8     double value() {
9         // your code goes here
10    }
11
12    void print() {
13        // your code goes here
14    }
15 };
16
17 Fraction add(Fraction x, Fraction y) {
18     // your code goes here
19 }
```

Question 4

Not answered

Mark 0.00 out of 10.00

[StudentStruct]

Sinh viên thực hiện các yêu cầu sau:

- Định nghĩa **cấu trúc** sinh viên **Student** gồm có:
 - Tên **name** kiểu dữ liệu string, điểm trung bình **gpa** kiểu số thực.
 - Hàm khởi tạo nhận vào tên và hàm khởi tạo nhận vào tên kèm điểm trung bình.
 - Hàm xếp loại bằng tốt nghiệp **string grade()**. Hàm trả về xếp loại bằng dựa vào điểm trung bình như sau:
 - "xuat sac" nếu $gpa \geq 3.6$.
 - "gioi" nếu $3.2 \leq gpa < 3.6$.
 - "kha" nếu $2.5 \leq gpa < 3.2$.
 - "khong du dieu kien" nếu $gpa < 2.5$.
 - Hàm **void print()** in ra thông tin sinh viên theo định dạng **tên điểm xếp_loại_bằng** (điểm làm tròn đến 1 chữ số sau dấu phẩy). In ký tự xuống dòng sau khi in xong.
- Định nghĩa hàm **Student get(int n)** như sau:
 - Đọc vào từ bàn phím n dòng mỗi dòng gồm một tên và một điểm trung bình của một sinh viên.
 - Trả về sinh viên có điểm cao nhất.

Lưu ý: Sinh viên chỉ thực hiện yêu cầu trên mà không viết hàm **main()**.

For example:

Test	Input	Result
Student a(stName); a.gpa= stGpa; cout << a.grade() << endl; a.print();	Viet 3.7	xuat sac Viet 3.7 xuat sac
get(3).print();	Viet 3.7 Tau 1.4 Nhat 2.8	Viet 3.7 xuat sac

Answer: (penalty regime: 0 %)

Reset answer

```
1 struct Student {
2     // your code goes here
3     // Cac bien thanh vien
4
5     // your code goes here
6     // Hai ham khoi tao
7
8     string grade() {
9         // your code goes here
10    }
11
12    void print() {
13        // your code goes here
14    }
15 };
16
17 Student get(int n) {
18     // your code goes here
19 }
```


Question 5

Not answered

Mark 0.00 out of 10.00

[Time 2]

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 thời gian ([chuẩn hóa](#) luôn dữ liệu đầu vào nếu `initSecond > 60`, `initMinute > 60` hoặc `initHour > 24`) và hàm `int checkArrivalTime(Time t1, Time t2)` kiểm tra xem t1 sớm hơn (trả về 1), t2 sớm hơn (trả về -1) hay bằng nhau (trả về 0).

Hoàn thiện chương trình dưới đây để nhập vào 2 mốc thời gian của 2 students.

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 ai là người đến sớm hơn, nếu đến cùng 1 giờ thì in ra dòng chữ "Equal".

For example:

Input	Result
7 5 30	1
7 15 40	
17 65 10	Equal
18 5 10	

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 };
13
14
15 int checkArrivalTime(Time t1, Time t2)
16 {
17     //your code here
18 }
19
20 int main()
21 {
22     int hour, minute, second;
23     cin >> hour >> minute >> second;
24     Time time1(hour,minute,second);
25     cin >> hour >> minute >> second;
26     Time time2(hour,minute,second);
27
28     if(checkArrivalTime(time1, time2) == 1) cout << "1";
29     else if(checkArrivalTime(time1, time2) == -1) cout << "2";
30     else cout << "Equal";
31
32     return 0;
33 }
34
```

[Back to Course](#)