

Trạng thái	Đã xong
Bắt đầu vào lúc	Thứ Ba, 8 tháng 4 2025, 10:53 AM
Kết thúc lúc	Thứ Ba, 8 tháng 4 2025, 11:53 AM
Thời gian thực hiện	1 giờ
Điểm	40,00/50,00
Điểm	8,00 trên 10,00 (80%)

## Câu hỏi 1

Đúng

Đạt điểm 10,00 trên 10,00

Viết phương thức **isClosed()** của lớp **Point** và cách gọi hàm này (nếu cần) để chương trình sau kiểm tra được hai điểm thuộc lớp **Point** có khoảng cách nhỏ hơn 1 hay không.

For example:

Input	Result
0 0 1 1	false

Answer: (penalty regime: 0 %)

Reset answer

```

1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5
6  const double MAX_DISTANT = 1.0;
7
8  class Point {
9      double x, y;
10     public:
11     Point (double _x, double _y) {
12         x = _x; y = _y;
13     }
14
15     bool isClosed(Point p1){
16         double distance=sqrt((p1.x-x)*(p1.x-x)+(p1.y-y)*(p1.y-y));
17         return distance<=1;
18     }
19
20 }
21 };
22
23 int main() {
24     double x1, y1, x2, y2;
25     cin >> x1 >> y1 >> x2 >> y2;
26
27     Point p1(x1, y1), p2(x2, y2);
28     cout << boolalpha << (p1.isClosed(p2));
29
30     return 0;
31 }
```

	Input	Expected	Got	
✓	0 0 1 1	false	false	✓
✓	0 0 1 0	true	true	✓
✓	0 0 0.5 0.5	true	true	✓
✓	0 0 2 0	false	false	✓

Passed all tests! ✓

**► SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)****Đúng**

Marks for this submission: 10,00/10,00.

**Câu hỏi 2**

Đúng

Đạt điểm 10,00 trên 10,00

Sinh viên thực hiện các yêu cầu sau:

- Định nghĩa cấu trúc hình chữ nhật **Rectangle** gồm có:
  - Độ dài hai cạnh **height**, **length** kiểu biến nguyên.
  - Hàm khởi tạo mặc định (gán hai cạnh bằng 0), và hàm khởi tạo nhận vào độ dài 2 cạnh.
  - Hàm tính và trả về chu vi hình chữ nhật **int getPerimeter()**.
  - Hàm **void print()** vẽ ra hình chữ nhật rỗng dùng dấu **\*** có chiều cao là **height** chiều rộng là **length**. In ký tự xuống dòng sau khi in xong.
- Định nghĩa hàm **int compare(Rectangle a, Rectangle b)** như sau:
  - Trả về -1 nếu a có chu vi nhỏ hơn b.
  - Trả về 0 nếu hai hình có chu vi bằng nhau.
  - Trả về 1 nếu a có chu vi lớn hơn b.

**For example:**

Test	Input	Result
Rectangle a; a.height = inputHeight; a.length = inputLength; cout << a.getPerimeter() << endl; a.print();	3 4	14 **** * * ****
Rectangle a(ha, la), b(hb, lb); cout << a.getPerimeter() << " " << b.getPerimeter() << endl; cout << compare(a, b);	2 5 3 4	14 14 0

**Answer:** (penalty regime: 0 %)

Reset answer

```

1 struct Rectangle {
2     // your code goes here
3     // Cac bien thanh vien
4     int height; int length;
5     // your code goes here
6     // Hai ham khoi tao
7     Rectangle (double _height, double _length){
8         height=_height;
9         length=_length;
10    }
11    Rectangle(): height(0), length(0){};
12    int getPerimeter() {
13        return 2*height+2*length;
14    }
15
16    void print() {
17        for(int i=0; i<height; i++){
18            for(int j=0; j<length; j++){
19                if(i==0 || i==height-1 || j==0 || j==length-1){
20                    cout<<"*";
21                }
22                else{
23                    cout<<" ";
24                }
25            }
26        }
27    }
28 }

```

```

26         cout<<endl;
27     }
28
29 }
30 };
31
32 int compare(Rectangle a, Rectangle b) {
33     if(a.getPerimeter()<b.getPerimeter()) return -1;
34     else if(a.getPerimeter()==b.getPerimeter()) return 0;
35     return 1;
36 }

```

	Test	Input	Expected	Got	
✓	Rectangle a; a.height = inputHeight; a.length = inputLength; cout << a.getPerimeter() << endl; a.print();	3 4	14 **** * * ****	14 **** * * ****	✓
✓	Rectangle a(ha, la), b(hb, lb); cout << a.getPerimeter() << " " << b.getPerimeter() << endl; cout << compare(a, b);	2 5 3 4	14 14 0	14 14 0	✓
✓	// test ham khoi tao Rectangle a; a.height = 2; a.length = 4; Rectangle b(2, 4); cout << a.length - b.length << endl; cout << a.height - b.height << endl; cout << a.length + a.height << endl;		0 0 6	0 0 6	✓
✓	// test ham tinh chu vi Rectangle a(3, 1), b(2, 4); cout << a.getPerimeter() << endl; cout << b.getPerimeter() << endl;		8 12	8 12	✓
✓	// test ham in Rectangle a(1, 1), b(2, 2), c(4, 3), d(4, 5); a.print(); b.print(); c.print(); d.print();		* ** ** *** * * * * *** ***** * * * * *****	* ** ** *** * * * * *** ***** * * * * *****	✓

	Test	Input	Expected	Got	
✓	<pre>// test ham so sanh Rectangle a(1, 1), b(2, 2), c(1, 3); cout &lt;&lt; compare(a, a) &lt;&lt; endl; cout &lt;&lt; compare(a, b) &lt;&lt; endl; cout &lt;&lt; compare(b, a) &lt;&lt; endl; cout &lt;&lt; compare(b, c) &lt;&lt; endl;</pre>		0 -1 1 0	0 -1 1 0	✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

**Câu hỏi 3**

Đúng

Đạt điểm 10,00 trên 10,00

Sinh viên thực hiện các yêu cầu sau:

- Định nghĩa cấu trúc số phức **Complex** gồm có:
  - Phần thực và phần ảo tương ứng là **a**, **b** kiểu số nguyên.
  - Hàm khởi tạo mặc định (gán hai phần bằng 0), và hàm khởi tạo nhận vào phần thực và phần ảo.
  - Hàm tính và trả về giá trị tuyệt đối của số phức **double abs()** với  $abs(a + bi) = \sqrt{a^2 + b^2}$ .
  - Hàm **void print()** in ra số phức dưới dạng  $a + bi$  hoặc  $a - bi$  tùy thuộc vào dấu của  $b$ . Nếu  $b = 1$  hoặc  $b = -1$  thì không in ra  $b$ . Nếu  $a$  hoặc  $b$  bằng 0 thì lược bỏ phần có chứa nó. Nếu cả  $a$  và  $b$  bằng 0 thì in ra 0. In thêm dấu xuống dòng ở cuối.
- Định nghĩa hàm **Complex add(Complex x, Complex y)** trả về tổng của hai số phức:
  - Cho biết  $(a_1 + b_1i) + (a_2 + b_2i) = (a_1 + a_2) + (b_1 + b_2)i$ .

For example:

Test	Input	Result
Complex x; x.a = inputA; x.b = inputB; x.print(); cout << fixed << setprecision(2) << x.abs();	3 4	3+4i 5.00
Complex x(xa, xb), y(ya, yb); x.print(); y.print(); add(x, y).print();	1 1 -2 -2	1+i -2-2i -1-i

**Answer:** (penalty regime: 0 %)

Reset answer

```

1 struct Complex {
2     // your code goes here
3     // Cac bien thanh vien
4     int a; int b;
5     // your code goes here
6     // Hai ham khoi tao
7     Complex (int _a, int _b){
8         a=_a;
9         b=_b;
10    }
11    Complex(): a(0), b(0){};
12    double abs() {
13        return sqrt(a*a+b*b);
14    }
15
16    void print() {
17        if(a==0 && b==0) cout<<"0\n";
18        else if(a==0){
19            cout<<b<<"i"<<endl;
20        }
21        else if(b==0){
22            cout<<a<<endl;
23        }
24        else if(b==1 || b==1){
25            cout<<a<<(b<0 ? "-" : "+")<<"i"<<endl;

```

```

26     }
27     else{
28         cout<<a<<(b<0 ? "" : "+")<<b<<"i"<<endl;
29     }
30 }
31 };
32
33 Complex add(Complex x, Complex y) {
34     int sumReal=x.a+y.a;
35     int sumImagine=x.b+y.b;
36     Complex res={sumReal, sumImagine};
37     return res;
38 }

```

	Test	Input	Expected	Got	
✓	Complex x; x.a = inputA; x.b = inputB; x.print(); cout << fixed << setprecision(2) << x.abs();	3 4	3+4i 5.00	3+4i 5.00	✓
✓	Complex x(xa, xb), y(ya, yb); x.print(); y.print(); add(x, y).print();	1 1 -2 -2	1+i -2-2i -1-i	1+i -2-2i -1-i	✓
✓	// test ham khoi tao Complex a; a.a = 1; a.b = 1; Complex b(2, 2); cout << a.a + b.a << endl; cout << a.b - b.b << endl; cout << a.a + a.b << endl;		3 -1 2	3 -1 2	✓
✓	// test ham tinh gia tri tuyet doi Complex a(1, 1), b(2, 2); cout << fixed << setprecision(2) << a.abs() << endl; cout << fixed << setprecision(1) << b.abs() << endl;		1.41 2.8	1.41 2.8	✓
✓	// test ham in Complex a(1, 1), b(0, -2), c(4, 0), d(4, -1), e(0, 0); a.print(); b.print(); c.print(); d.print(); e.print();		1+i -2i 4 4-i 0	1+i -2i 4 4-i 0	✓
✓	// test ham cong Complex a(1, 1), b(2, 2), c(1, -3); add(a, b).print(); add(b, c).print(); add(a, c).print(); add(a, a).print();		3+3i 3-i 2-2i 2+2i	3+3i 3-i 2-2i 2+2i	✓



Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

**Câu hỏi 4**

Đúng

Đạt điểm 10,00 trên 10,00

Sinh viên thực hiện các yêu cầu sau:

- Định nghĩa cấu trúc thời gian **Time** gồm có:
  - Giờ, phút, giây tương ứng là **h**, **m**, **s** kiểu số nguyên (luôn dương).
  - Hàm khởi tạo mặc định (gán giờ, phút, giây bằng 0), và hàm khởi tạo nhận vào ba số giờ, phút, giây.
  - Hàm tính số giây **int second()** tương ứng với thời gian.
  - Hàm **void print()** in ra thời gian theo định dạng **hh : mm : ss** (ví dụ 08 : 20 : 10). In thêm dấu xuống dòng ở cuối.
- Định nghĩa hàm **Time normalize(int h, int m, int s)** chuẩn hoá thời gian:
  - Nhận vào 3 biến nguyên.
  - Chuẩn hoá nếu **m**, **s** lớn hơn 59 (ví dụ 02 : 59 : 60 chuẩn hoá thành 03 : 00 : 00).
  - Trả về biến thời gian đã chuẩn hoá.

**For example:**

Test	Input	Result
Time x; x.h = inputH; x.m = inputM; x.s = inputS; x.print(); cout << x.second();	1 20 3	01:20:03 4803
normalize(inputH, inputM, inputS).print();	9 59 60	10:00:00

**Answer:** (penalty regime: 0 %)

Reset answer

```

1 struct Time {
2     // your code goes here
3     int h, m, s;
4     // your code goes here
5     Time (int _h, int _m, int _s){
6         h=_h;
7         m=_m;
8         s=_s;
9     }
10    Time(): h(0), m(0), s(0){};
11
12    int second() {
13        return h*60*60+m*60+s;
14    }
15
16    void print() {
17        if(s>=60){
18            m+=s/60;s%=60;
19        }
20        if(m>=60){
21            h+=m/60;
22            m%=60;
23        }
24
25        cout<<(h>=10 ? "" : "0")<<h<<":";
26        cout<<(m>=10 ? "" : "0")<<m<<":";
27        cout<<(s>=10 ? "" : "0")<<s<<endl;
28    }
29 }

```

```

40     }
29 };
30
31 Time normalize(int h, int m, int s) {
32     if(s>=60){
33         m+=s/60;s%=60;
34     }
35     if(m>=60){
36         h+=m/60;
37         m%=60;
38     }
39     Time res={h,m,s};
40     return res;
41 }

```

	Test	Input	Expected	Got	
✓	Time x; x.h = inputH; x.m = inputM; x.s = inputS; x.print(); cout << x.second();	1 20 3	01:20:03 4803	01:20:03 4803	✓
✓	normalize(inputH, inputM, inputS).print();	9 59 60	10:00:00	10:00:00	✓
✓	// test ham khoi tao Time a; a.h = 1; a.m = 1; a.s = 1; Time b(2, 2, 2); cout << a.h + a.m + a.s << endl; cout << a.h - b.m << endl; cout << a.m + b.s << endl;		3 -1 3	3 -1 3	✓
✓	// test ham tinh giay Time a(1, 1, 1), b(0, 0, 0); cout << a.second() << endl; cout << b.second() << endl;		3661 0	3661 0	✓
✓	// test ham in Time a(1, 1, 1), b(12, 12, 12), c(4, 0, 1), d(3, 0, 0), e(0, 0, 0); a.print(); b.print(); c.print(); d.print(); e.print();		01:01:01 12:12:12 04:00:01 03:00:00 00:00:00	01:01:01 12:12:12 04:00:01 03:00:00 00:00:00	✓
✓	// test ham chuan hoa normalize(1, 1, 1).print(); normalize(10, 10, 60).print(); normalize(10, 10, 3600).print(); normalize(10, 10, 3000).print();		01:01:01 10:11:00 11:10:00 11:00:00	01:01:01 10:11:00 11:10:00 11:00:00	✓

Passed all tests! ✓

Đúng

Marks for this submission: 10,00/10,00.

Câu hỏi 5

Sai

Đạt điểm 0,00 trên 10,00

triển khai lại phiên bản đơn giản của lớp std::vector C++. Mục tiêu của bạn là tạo một lớp Vector dựa trên mẫu mô phỏng hành vi của vector chuẩn, bao gồm thay đổi kích thước động và các thao tác cơ bản.

For example:

Test	Input	Result
<pre>MyVector vec;  // Push elements vec.push_back(10); vec.push_back(20); vec.push_back(30);  std::cout &lt;&lt; "Size: " &lt;&lt; vec.size() &lt;&lt; std::endl;  // Access elements std::cout &lt;&lt; "Element at index 1: " &lt;&lt; vec[1] &lt;&lt; std::endl;  // Pop last element vec.pop_back(); std::cout &lt;&lt; "After pop_back, Size: " &lt;&lt; vec.size() &lt;&lt; std::endl;  // Clear all elements vec.clear(); std::cout &lt;&lt; "After clear, Size: " &lt;&lt; vec.size() &lt;&lt; std::endl;  // Add new elements vec.push_back(40); vec.push_back(50);  // Use iterator to print elements std::cout &lt;&lt; "Elements in vector: "; for (auto it = vec.begin(); it != vec.end(); ++it) {     std::cout &lt;&lt; *it &lt;&lt; " "; }  std::cout &lt;&lt; std::endl;</pre>	1	<pre>Size: 3 Element at index 1: 20 After pop_back, Size: 2 After clear, Size: 0 Elements in vector: 40 50</pre>

Answer: (penalty regime: 10, 20, ... %)

Reset answer

```
1 class MyVector {
2 private:
3     int* arr;           // Pointer to the array of elements
4     size_t sz;          // Number of elements in the vector
5     size_t cap;         // Capacity of the vector
6
7 public:
8     MyVector(): arr(nullptr), sz(0), cap(0){};
9     ~MyVector(){
10         delete[] arr;
```