Problem: Find the midpoint of a given line.  
  
Sample:

|  |  |
| --- | --- |
| Input | Output |
| 2/1 2/1 6/1 6/1 0/1 5/2 0/1 3/2 | 4/1 4/1  0/1 4/2  (no need to show in reduced format) |

Steps:

1. Define a class Fraction with two public attribute **NUMERATOR** and **DENOMINATOR.**
2. Define a function Fraction addFraction(Fraction a) that will return another fraction that is sum of fraction a and the caller Fraction.

(3/5, 5/10) → (3\*10+5\*5)/50 → 55/50 [no need to think of reduced form]

1. Define a function Fraction divFraction(int n) that will return the fraction after dividing the caller Fraction by n.  
     
   (2/4, 2) → (2/8) [no need to think of reduced form]

1. Define a class Point with two private attribute x & y, where both are Fraction.
2. Define a class Line with two public attribute start & end, where both are Point.
3. Define a function Point midPoint() that will return mid point of the caller line. Use the functions you defined in step 2 & 3.

Problem: Find If a point in a line or not.

1. Define a friend function bool contains(Line l, Point p), check if Point p contains l or not.

Problem: Find the Intersection of two lines.

1. Define a function bool intersection(Line l, Point& p), that check if line l intersect with caller fraction or not. If intersects p contains the intersecting point.

|  |
| --- |
| **#include<iostream>**  **using namespace std;**  **class Fraction {**  **private:**  **public:**  **int n, d;**  **Fraction() {**  **}**  **Fraction(int \_n, int \_d) {**  **n = \_n;**  **d = \_d;**  **}**  **void print() {**  **cout << n<<"/" << d<<endl;**  **}**  **void scan() {**  **char dummy;**  **cin >> n >> dummy >> d;**  **}**  **Fraction addFraction(Fraction a) {**  **Fraction ret;**  **ret.n = n\*a.d + a.n\*d;**  **ret.d = d \* a.d;**  **return ret;**  **}**  **Fraction divFraction(int n) {**  **Fraction ret;**  **ret.n = this->n;**  **ret.d = d \* n;**  **return ret;**  **}**  **};**  **class Point {**  **private:**  **// make x, y private**  **public:**  **Fraction x, y;**  **Point() {}**  **Point(Fraction x, Fraction y) {**  **this->x = x;**  **this->y = y;**  **}**  **void print() {**  **cout <<"x: ";**  **x.print();**  **cout <<"y: ";**  **y.print();**  **}**  **Fraction getX() {**  **return x;**  **}**  **Fraction getY() {**  **return y;**  **}**  **};**  **class Line {**  **public:**  **Point start, end;**  **Line(){}**  **Line(Point start, Point end) {**  **this->start = start;**  **this->end = end;**  **}**  **void print() {**  **cout <<"start: \n";**  **start.print();**  **cout <<"end: \n";**  **end.print();**  **}**  **Point midPoint() {**  **Point mid;**  **mid.x = start.x.addFraction(end.x); /// mid.x = (start.x+end.x);**  **mid.x = mid.x.divFraction(2); /// mid.x = mid.x/2;**  **mid.y = start.y.addFraction(end.y);**  **mid.y = mid.y.divFraction(2);**  **return mid;**  **}**  **};**  **int main() {**  **Fraction f1(2,1); // start.x**  **Fraction f2(2,1);**  **Fraction f3(6,1);**  **Fraction f4(6,1);**  **Point a( f1, f2 );**  **Point b( f3, f4 );**  **Line l(a, b);**  **l.print();**  **Point mid = l.midPoint();**  **cout << endl;**  **mid.print();**  **// cout << l.end.getX().d <<endl;**  **// cout << l.end.getY().n <<endl;**  **}** |

**You may also do it in the following way: Here addFraction and divFraction are not class method of Fraction class. So they take two parameters and do the operation on them and return the desired fraction.**

|  |
| --- |
| **#include<iostream>**  **using namespace std;**  **class Fraction {**  **private:**  **public:**  **int n, d;**  **Fraction() {**  **}**  **Fraction(int \_n, int \_d) {**  **n = \_n;**  **d = \_d;**  **}**  **void print() {**  **cout << n<<"/" << d<<endl;**  **}**  **void scan() {**  **char dummy;**  **cin >> n >> dummy >> d;**  **}**  **};**  **Fraction addFraction(Fraction a, Fraction b) {**  **Fraction ret;**  **ret.n = b.n\*a.d + a.n\*b.d;**  **ret.d = b.d \* a.d;**  **return ret;**  **}**  **Fraction divFraction(Fraction a, int n) {**  **Fraction ret;**  **ret.n = a.n;**  **ret.d = a.d \* n;**  **return ret;**  **}**  **class Point {**  **private:**  **// make x, y private**  **public:**  **Fraction x, y;**  **Point() {}**  **Point(Fraction x, Fraction y) {**  **this->x = x;**  **this->y = y;**  **}**  **void print() {**  **cout <<"x: ";**  **x.print();**  **cout <<"y: ";**  **y.print();**  **}**  **Fraction getX() {**  **return x;**  **}**  **Fraction getY() {**  **return y;**  **}**  **};**  **class Line {**  **public:**  **Point start, end;**  **Line(){}**  **Line(Point start, Point end) {**  **this->start = start;**  **this->end = end;**  **}**  **void print() {**  **cout <<"start: \n";**  **start.print();**  **cout <<"end: \n";**  **end.print();**  **}**  **Point midPoint() {**  **Point mid;**  **mid.x = addFraction(start.x, end.x); /// mid.x = (start.x+end.x);**  **mid.x = divFraction(mid.x, 2); /// mid.x = mid.x/2;**  **mid.y = addFraction(start.y, end.y);**  **mid.y = divFraction(mid.y, 2);**  **return mid;**  **}**  **};**  **int main() {**  **Fraction f1(2,1); // start.x**  **Fraction f2(2,1);**  **Fraction f3(6,1);**  **Fraction f4(6,1);**  **Point a( f1, f2 );**  **Point b( f3, f4 );**  **Line l(a, b);**  **l.print();**  **Point mid = l.midPoint();**  **cout << endl;**  **mid.print();**  **// cout << l.end.getX().d <<endl;**  **// cout << l.end.getY().n <<endl;**  **}** |