Name : Mukul Kumar

Enrollment number : 201B162

Batch : B5

Subject : OOPL

Lab assignment number : 3

Semester : 2

1. Write C++ Program to swap two variable using reference variables.

Code:

#include <iostream>

using std::cin;

using std::cout;

using std::endl;

using std::string;

void swap(string &, string &);

int main()

{

string a, b;

string &ref\_a = a;

string &ref\_b = b;

cout << "Input variable a" << endl;

cin >> a;

cout << "Input variable b" << endl;

cin >> b;

swap(ref\_a, ref\_b);

cout << "Now a variable is " << a << "\n"

<< "And b variable is " << b << endl;

return 0;

}

void swap(string &ref\_a, string &ref\_b)

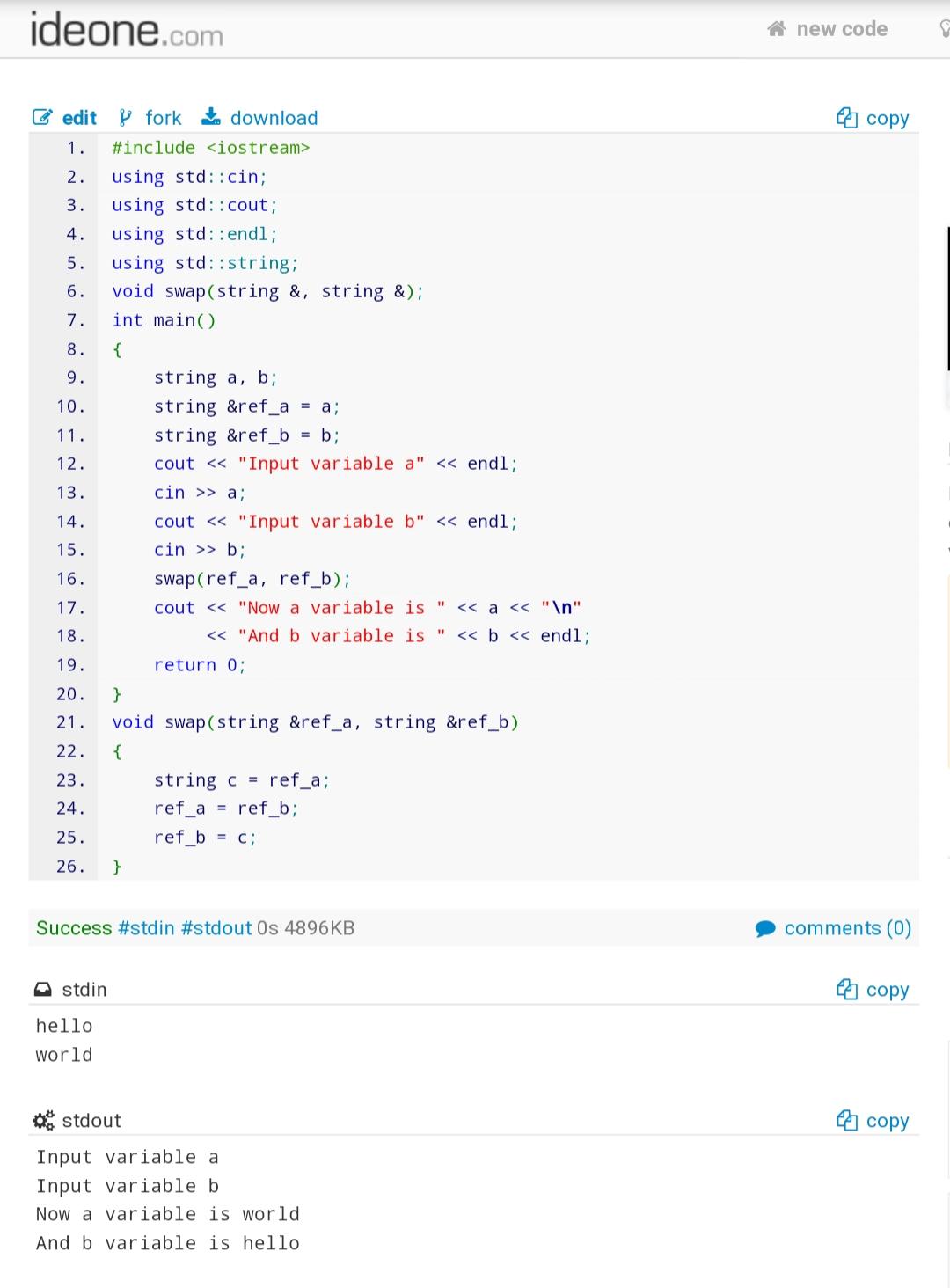
{

string c = ref\_a;

ref\_a = ref\_b;

ref\_b = c;

}



2. Write a function that finds the minimum and the maximum value in an array of N integers.

Inputs to the function are the array of integers, an integer variable containing the length of the

array and references to integer variables that will contain the minimum and the maximum

values. The function prototype is:

void minmax (int array[], int length, int& min, int & max);

Code:

#include <iostream>

using std::cin;

using std::cout;

using std::endl;

void minmax(int [],int,int &,int &);

int main()

{

int n;

cin>>n;

int arr[n],max,min;

int &ref\_max = max;

int &ref\_min = min;

for (int i = 0; i<n;i++)

{

cin>> arr[i];

}

minmax(arr,n,ref\_max,ref\_min);

cout << "max is : " << max << "\nmin is " << min << endl;

}

void minmax(int arr[],int n,int& max,int &min)

{

max = arr[0];

min = arr[0];

for (int i = 1 ; i < n;i++)

{

if (max<arr[i])

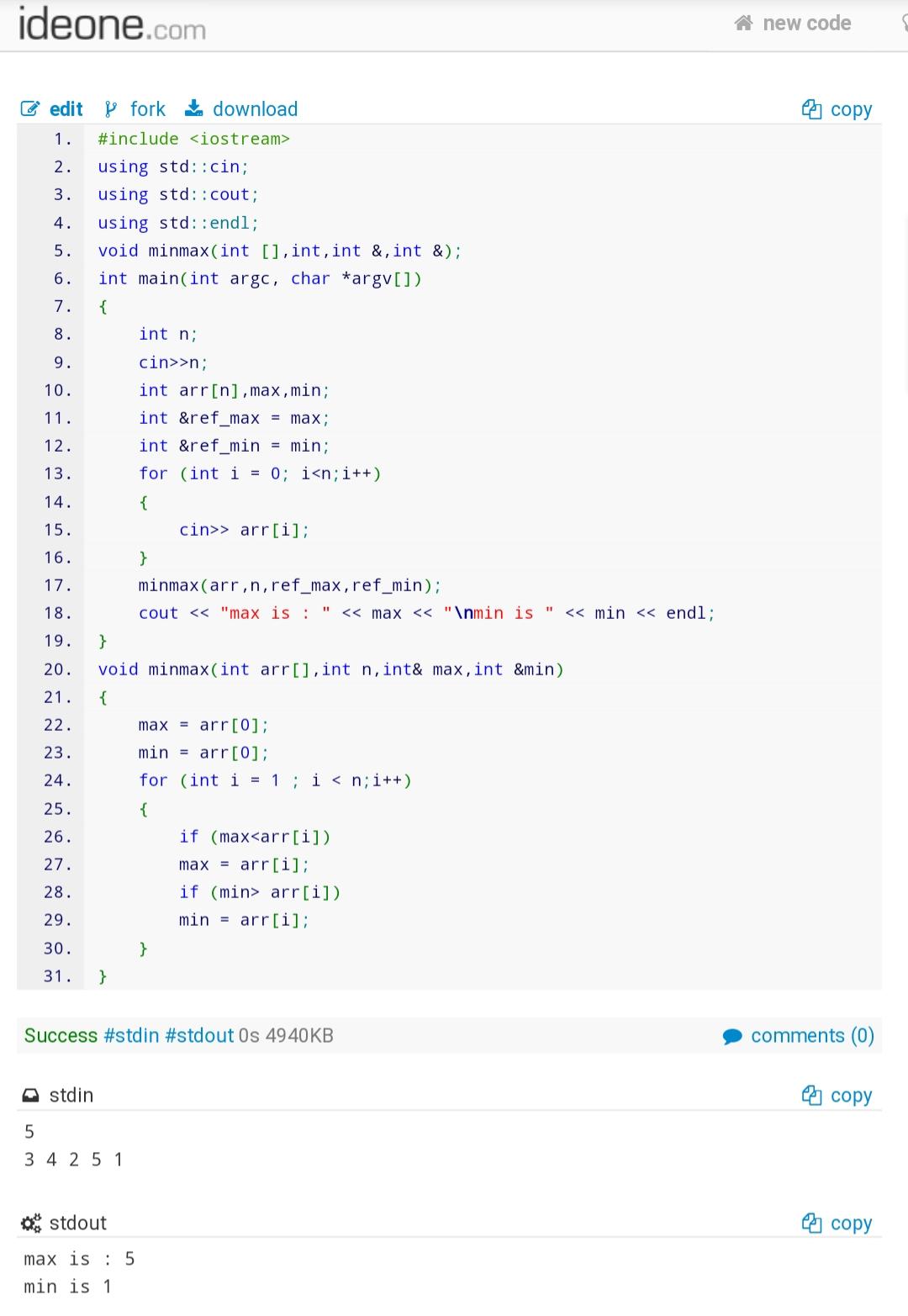
max = arr[i];

if (min> arr[i])

min = arr[i];

}

}



3. Create a four-function calculator for fractions. Here are the formulas for the four arithmetic

operations applied to fractions:

Addition: a/b + c/d = (a\*d + b\*c) / (b\*d)

Subtraction: a/b - c/d = (a\*d - b\*c) / (b\*d)

Multiplication: a/b \* c/d = (a\*c) / (b\*d)

Division: a/b / c/d = (a\*d) / (b\*c)

The user should type the first fraction (two values a and b), an operator, and a second fraction (two

values c and d). The program should then display the results in fraction ie. (Numerator/denominator)

Code:

#include <iostream>

using std::cin;

using std::cout;

using std::endl;

int gcd(int, int);

void simplify(int &, int &);

void add(int, int, int, int, int &, int &);

void sub(int, int, int, int, int &, int &);

void mul(int, int, int, int, int &, int &);

void div(int, int, int, int, int &, int &);

int main()

{

int a, b, c, d, numer, denom;

int &ref\_numer = numer;

int &ref\_denom = denom;

char op;

cout << "input (a/b) as a and b : ";

cin >> a >> b;

cout << "input operator : ";

cin >> op;

cout << "input (c/d) as c and d : ";

cin >> c >> d;

switch (op)

{

case '+':

if (b == 0 || d==0){

cout<<"division by 0 not possible"<<endl;

break;

}

add(a, b, c, d, ref\_numer, ref\_denom);

cout << "sum is\n"

<< numer << "\n----\n" << denom << endl;

break;

case '-':

if (b == 0 || d==0){

cout<<"division by 0 not possible"<<endl;

break;

}

sub(a, b, c, d, ref\_numer, ref\_denom);

cout << "sub is\n"

<< numer << "\n----\n" << denom << endl;

break;

case '\*':

if (b == 0 || d==0){

cout<<"division by 0 not possible"<<endl;

break;

}

mul(a, b, c, d, ref\_numer, ref\_denom);

cout << "mul is\n"

<< numer << "\n----\n" << denom << endl;

break;

case '/':

if (b == 0 || d==0){

cout<<"division by 0 not possible"<<endl;

break;

}

div(a, b, c, d, ref\_numer, ref\_denom);

cout << "div is\n"

<< numer << "\n----\n" << denom << endl;

break;

}

}

void add(int a, int b, int c, int d, int &numer, int &denom)

{

numer = (a \* d) + (b \* c);

denom = (b \* d);

simplify(numer, denom);

}

void sub(int a, int b, int c, int d, int &numer, int &denom)

{

numer = (a \* d) - (b \* c);

denom = (b \* d);

simplify(numer, denom);

}

void mul(int a, int b, int c, int d, int &numer, int &denom)

{

numer = (a \* d) \* (b \* c);

denom = (b \* d);

simplify(numer, denom);

}

void div(int a, int b, int c, int d, int &numer, int &denom)

{

numer = (a \* d) / (b \* c);

denom = (b \* d);

simplify(numer, denom);

}

void simplify(int &numer, int &denom)

{

int hcf = gcd(numer,denom);

numer = numer/hcf;

denom = denom/hcf;

}

int gcd(int a, int b)

{

if (a == 0)

return b;

if (b == 0)

return a;

if (a == b)

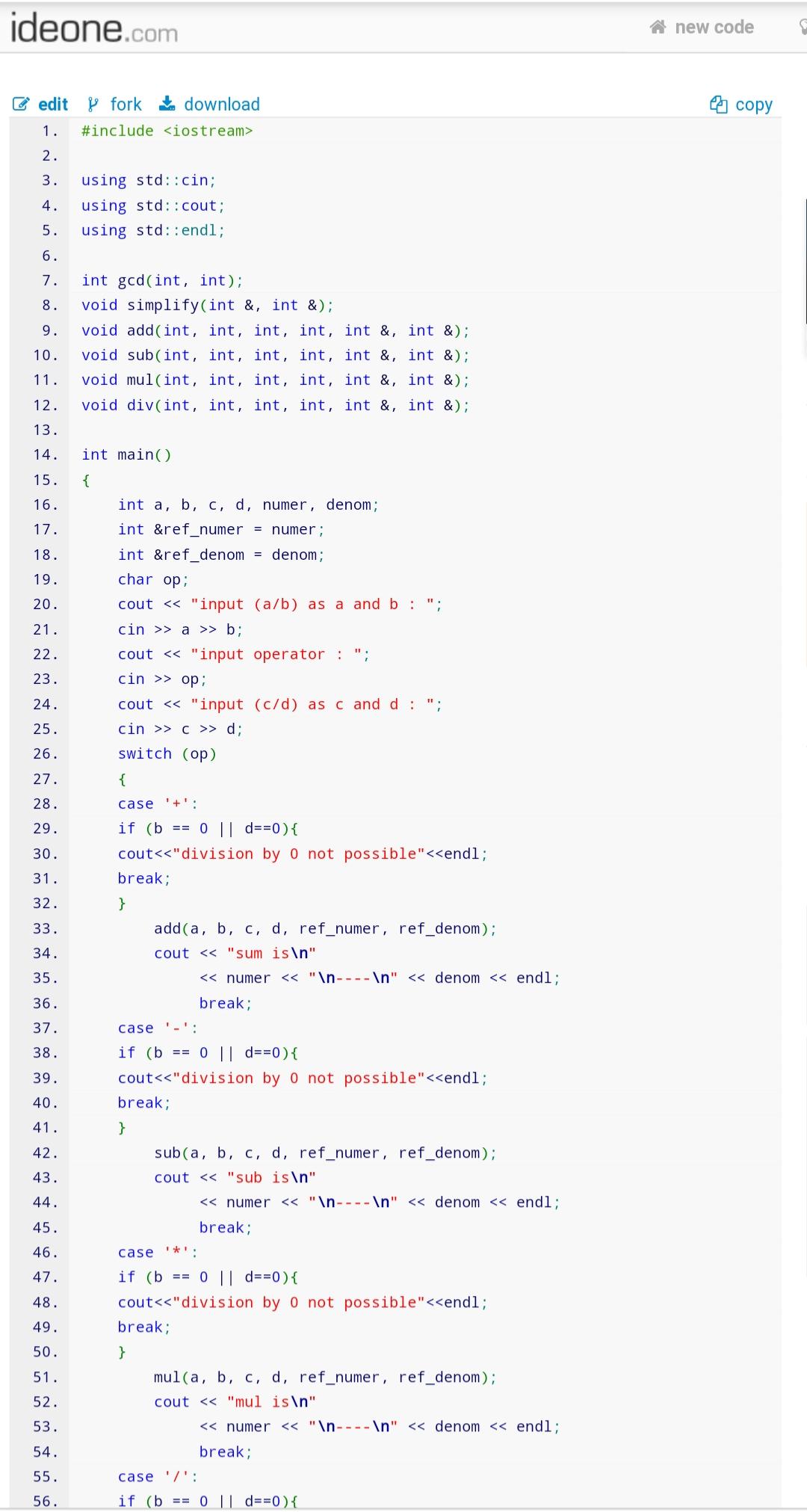
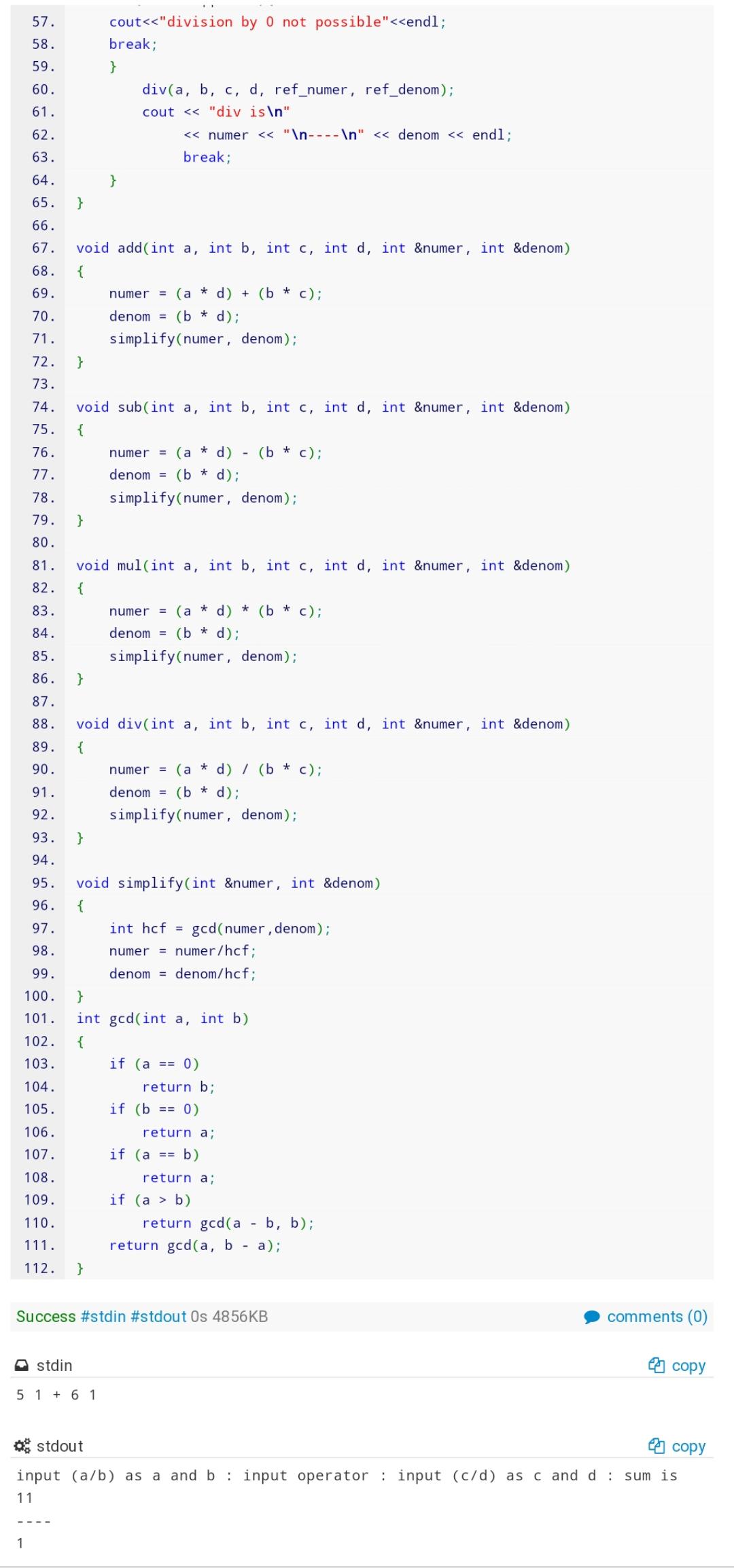
return a;

if (a > b)

return gcd(a - b, b);

return gcd(a, b - a);

}



4. Create a class rectangle with attributes length and width. Provide member functions that calculate the perimeter and area of the rectangle. Provide member functions to get the values from users and display the values of member variables. Write a program to test the class.

Code:

#include <iostream>

using std::cin;

using std::cout;

using std::endl;

class rectangle

{

int length;

int width;

public:

void get\_parameters()

{

length = 1;

width = 1;

}

void get\_parameters(int length, int width)

{

this->length = length;

this->width = width;

}

int area();

int perimeter();

};

int rectangle::area()

{

return length \* width;

}

int rectangle::perimeter()

{

return 2 \* (length + width);

}

int main()

{

int x, y;

rectangle r1;

cout << "Input length and width of ractangle" << endl;

cin >> x >> y;

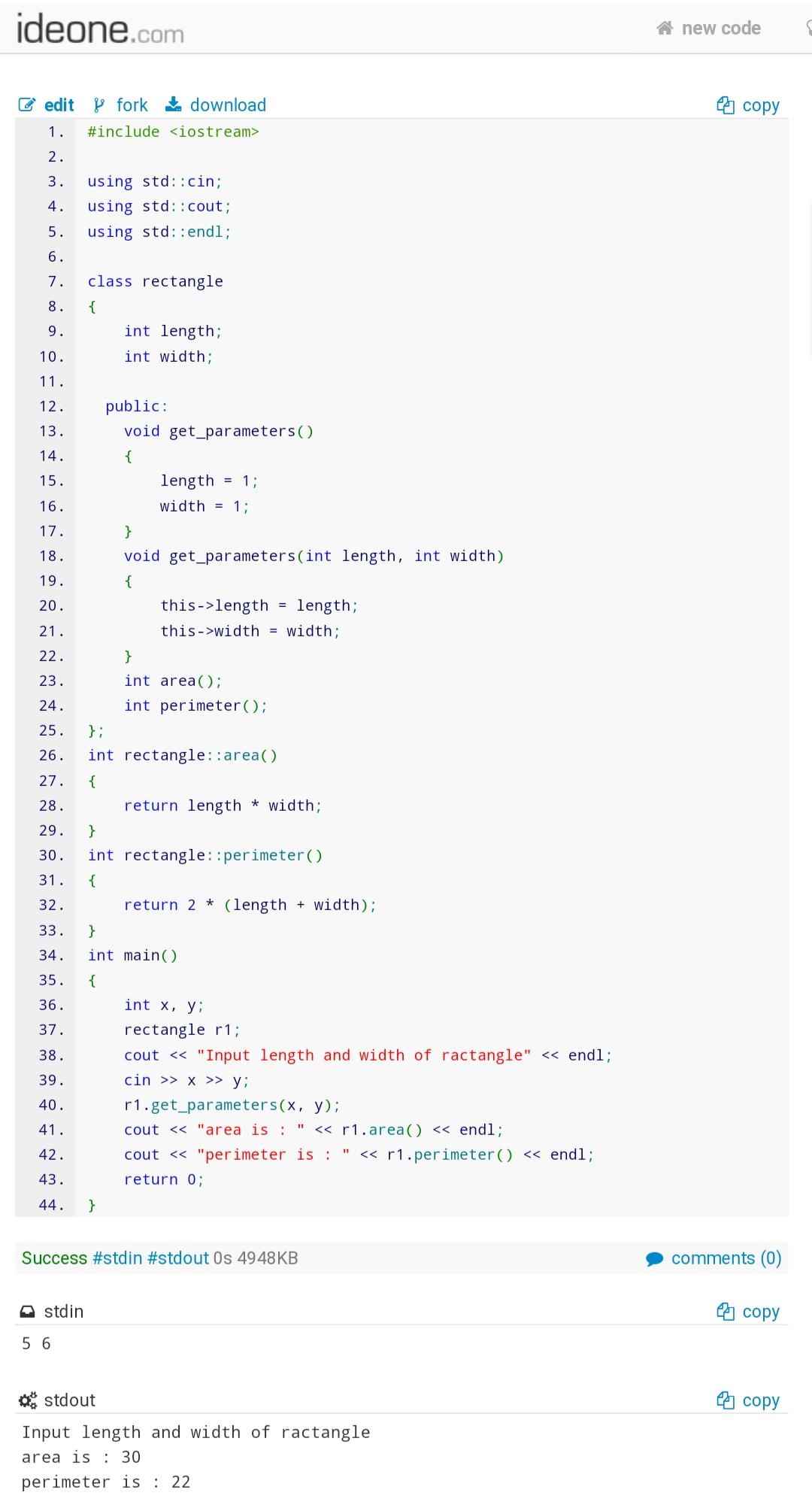
r1.get\_parameters(x, y);

cout << "area is : " << r1.area() << endl;

cout << "perimeter is : " << r1.perimeter() << endl;

return 0;

}



5. Write a function that accepts two arguments: a string name of a movie and an integer running time in minutes. Provide a default value for the minutes so that if you call the function without an integer argument, the minutes default to 90. Write a main() function that proves you can call the function with a string argument alone as well as with a string and an integer.

Code:

#include <iostream>

#include <stdlib.h>

#include <string>

using std::cin;

using std::cout;

using std::endl;

using std::string;

void movie(string name, int time = 90)

{

cout << "Name of movie is " << name << " with running time of " << time << endl;

}

int main()

{

string moviename;

int time;

cout << "Enter name of movie" << endl;

cin >> moviename;

cout << "Enter movie running time" << endl;

cin >> time;

cout << "All arguments are given to function" << endl;

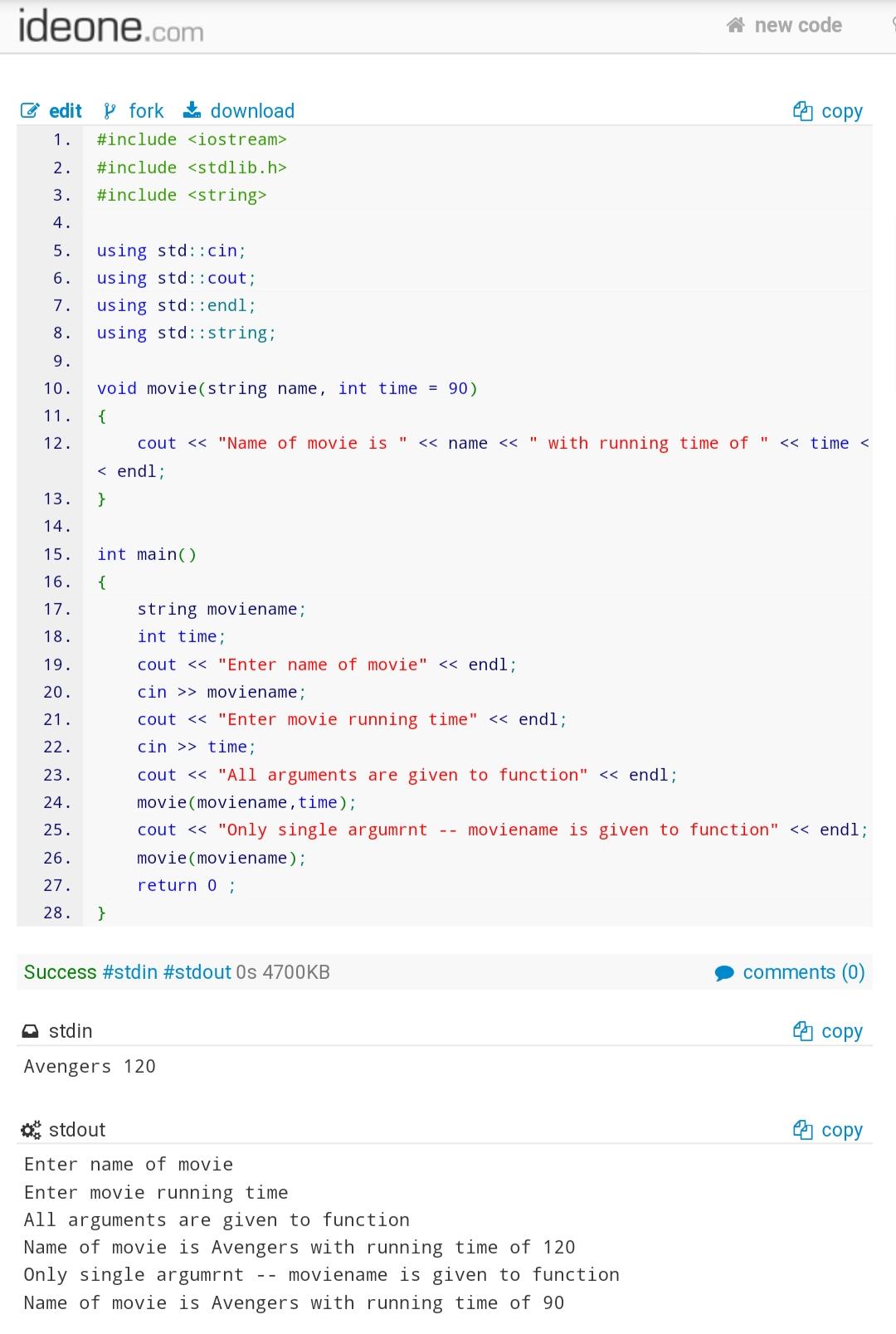
movie(moviename,time);

cout << "Only single argumrnt -- moviename is given to function" << endl;

movie(moviename);

return 0 ;

}



6. Create a class named Shirt that has the public data members collarsize and sleeveLength. Create a class named Pants that has the public data members waistSize and inSeam. Write a program that declares one object of each type Shirt and Pants and assigns values to the objects’ data fields. Write two overloaded functions named displayClothingFacts(). One version of the function takes a Shirt object as an argument; the other version takes a Pants object. Each version displays the facts about the piece of clothing. Your main() function should demonstrate that you can call displayClothingFacts() with either type of clothing.

Code:

#include <iostream>

using std::cin;

using std::cout;

using std::endl;

class Shirt

{

public:

int collarsize;

int sleeveLength;

};

class Pants

{

public:

int waistSize;

int inSeam;

};

void displayClothingFacts(Shirt);

void displayClothingFacts(Pants);

int main()

{

Shirt s1 = {5,10};

Pants p1 = {6,12};

displayClothingFacts(s1);

displayClothingFacts(p1);

}

void displayClothingFacts(Shirt s)

{

cout << "collorsize : " << s.collarsize << "\nsleeveLength : " << s.sleeveLength << endl;

}

void displayClothingFacts(Pants p)

{

cout << "waistSize : " << p.waistSize << "\ninSteam : " << p.inSeam << endl;

}



7. Define a class named Movie. Include private fields for the title, year, and name of the director. Include three public functions with the prototypes void Movie::setTitle(string); , void Movie::setYear(int); and void Movie::setDirector(string);. Include another function that displays all the information about a Movie. Write a main() function that declares a movie object named myFavoriteMovie. Set and display the object’s fields.

Code:

#include <iostream>

using std::cin;

using std::cout;

using std::endl;

using std::string;

class Movie

{

string title;

string name\_of\_director;

int year;

public:

void setTitle(string);

void setYear(int);

void setDirector(string);

void displayInformation();

};

void Movie::setTitle(string title)

{

this->title = title;

}

void Movie::setYear(int year)

{

this->year = year;

}

void Movie::setDirector(string Director)

{

name\_of\_director = Director;

}

void Movie::displayInformation()

{

cout << "Movie title : " << title << endl;

cout << "Movie release year : " << year << endl;

cout << "Movie Director name : " << name\_of\_director << endl;

}

int main()

{

Movie m1;

string title, director;

int year;

cout << "Input movie title" << endl;

cin >> title;

cout << "Input movie year" << endl;

cin >> year;

cout << "Input movie director" << endl;

cin >> director;

m1.setYear(year);

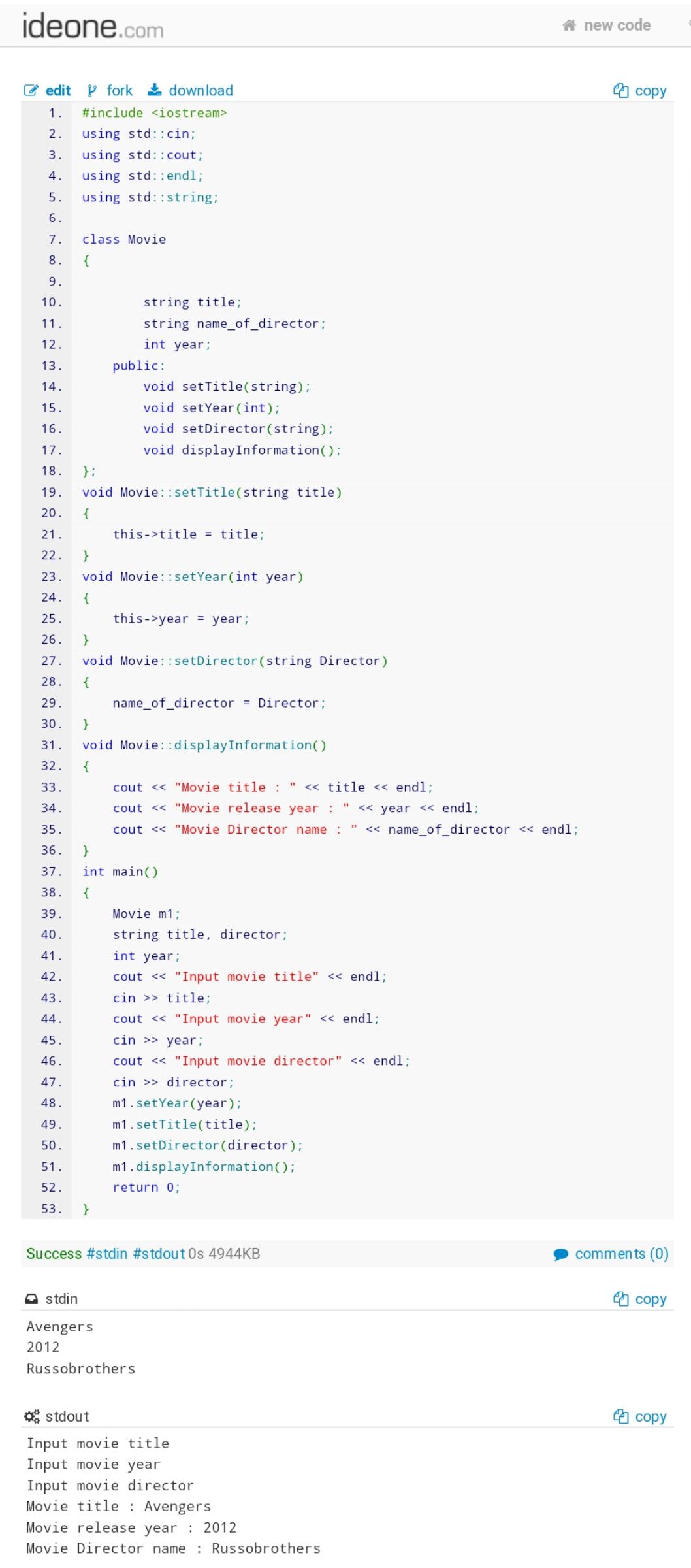
m1.setTitle(title);

m1.setDirector(director);

m1.displayInformation();

return 0;

}



8. Write a class definition for an order class for a nightclub that contains a table number, a server’s name, and the number of patrons at the table. Include a private static data member for the table minimum charge, which is $4.75. Write a main() function that declares no object of order class type, but uses a static member function to display the table minimum charge.

Code:

#include <iostream>

using std::cin;

using std::cout;

using std::endl;

using std::string;

class order

{

static float minimum\_charge;

int table\_number;

string server\_name;

int number\_of\_patrons;

public:

void getdata(){

cin >> table\_number;

getline(cin, server\_name);

cin >> number\_of\_patrons;

}

static void display\_minimum\_charge()

{

cout << minimum\_charge << endl;

}

};

float order::minimum\_charge = 4.75;

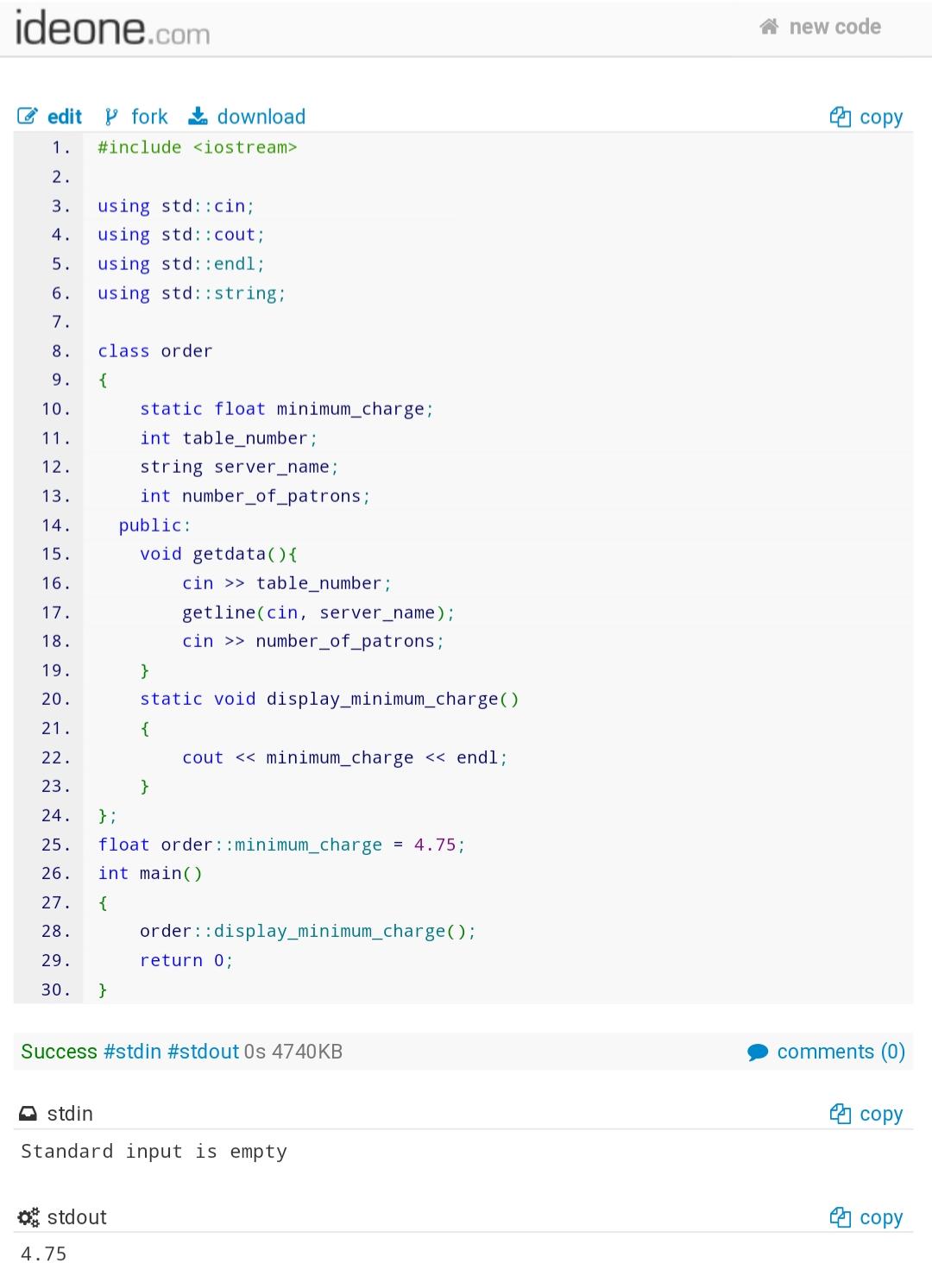
int main()

{

order::display\_minimum\_charge();

return 0;

}



9. Write a c++ program to find the highest occurring digit in prime numbers in a given range.

Given a range L to R, the task is to find the highest occurring digit in prime numbers lie between L and R (both inclusive). If multiple digits have same highest frequency print the largest of them. If no prime number occurs between L and R, output -1.

Examples:

Input : L = 1 and R = 20.

Output : 1

Prime number between 1 and 20 are 2, 3, 5, 7, 11, 13, 17, 19.

1 occur maximum i.e 5 times among 0 to 9.

Code:

#include <iostream>

#include <math.h>

using std::cin;

using std::cout;

using std::endl;

int \*primearray(int, int, int &);

int number\_counter(int \*, int);

int main()

{

int start, end, length;

int &ref\_length = length;

cin >> start >> end;

cout << number\_counter(primearray(start, end, ref\_length), length) << endl;

}

int \*primearray(int start, int end, int &length)

{

int \*ptr;

length = 0;

ptr = (int \*)malloc(length \* sizeof(int));

for (int I = start; I <= end; i++)

{

int flag = 1;

for (int j = 2; j <= sqrt(i); j++)

{

if (I % j == 0)

{

flag = 0;

break;

}

}

if (flag == 1 && I != 1)

{

length++;

\*(ptr + length – 1) = I;

ptr = (int \*)realloc(ptr, (length + 1) \* sizeof(int));

}

if (length == 0)

\*ptr = -1;

}

return ptr;

}

int number\_counter(int \*arr, int length)

{

int backup\_array[length];

for (int I = 0; I < length; i++)

{

backup\_array[i] = \*(arr + i);

}

int array[10] = {0};

for (int I = 0; I < length; i++)

{

while (\*(arr + i) > 0)

{

int rem;

rem = \*(arr + i) % 10;

\*(arr + i) = \*(arr + i) / 10;

array[rem]++;

}

}

int max = arr[0], array\_index = 0;

if (\*arr == -1)

array\_index = -1;

else

{

for (int I = 1; I < 10; i++)

{

if (max <= array[i])

{

max = array[i];

array\_index = I;

}

}

}

return array\_index;

}

