al Marks: 187.0

Question 1 Max. Marks 6.00

In Java, which of the following statements about *Interface* are true:

1. An interface can extend multiple interfaces.
2. An interface cannot extend any class.
3. Interface methods can define an implementation by using the default methods.
4. A static method in a class does not hide or override the static method in the interface that it implements.

- 1, 2, and 3
- 1, 2, and 4
- 1, 3, and 4
- All of these

Submit

Reset Answer

Next Question

Question 2 Max. Marks 6.00

If you are transforming an **ER model** to a **relational model**, then which of the following statements are true:

1. Entities in the ER Model are transformed into tables.
2. Attributes of the entities are converted to columns of the table.
3. For any relationship of a weak entity, if the primary key of any other entity is included in a table, then the foreign key constraint must be defined.
4. The relationship between entities is also converted into a table with proper foreign key constraints.

- 1, 2, and 3
- 1 and 2
- 1, 2, and 4
- All of these

Submit Reset Answer

Next Question

MSTC- API Development

01:06:15 left

arulsuju@gmail.com Help End Test

Total Marks: 187.0

Questions

he following + 6.0
t interface a...

ansforming an ER + 6.0
elational model, t...

oot, which of the + 4.0
ode is used to disa...

hese statements + 6.0
n Factory in Spring is tr...

these best completes the + 6.0
the following RE...

he return statement in the + 4.0
Java code. ...

the output of the + 4.0
g RESTful web service

Question 3

Max. Marks 4.00

In Spring Boot, which of the following codes is used to disable specific auto-configuration classes?

- @Configuration @EnableAutoConfiguration(exclude={DataSourceAutoConfiguration.class}) public class CustomConfiguration { }
- @Configuration @DisableAutoConfiguration(exclude={DataSourceAutoConfiguration.class}) public class CustomConfiguration { }
- @Config @EnableConfiguration(exclude={DataSourceAutoConfiguration.class}) public class CustomConfiguration { }
- @Config @AutoConfiguration(exclude={DataSourceAutoConfiguration.class}) public class CustomConfiguration { }

Submit

Reset Answer

Next Question

MSTC- API Development

01:06:08 left

arulsuju@gmail.com Help End

Total Marks: 187.0

stions

the following + 6.0
ut Interface a...

nsforming an ER + 6.0
elational model, t...

ot, which of the + 4.0
des is used to disa...

ese statements + 6.0
Factory in Spring is tr...

hese best completes the + 6.0
he following RE...

e return statement in the + 4.0
java code: ...

he output of the + 4.0
RESTful web service

Question 4

Max. Marks 6.00

Which of these statements about **Bean Factory** in Spring is true?

- A BeanFactory maintains a registry of different beans and their dependencies.
- A BeanFactory enables you to read bean definitions and access them using the bean factory.
- A BeanFactory creates and reads some bean definitions in the XML format as: Resource definition1 = new FileSystemResource("beans-def.xml"); BeanFactory factory = new XmlBeanFactory(definition1)
- All of these

Submit

Reset Answe

MSTC- API Development

01:06:02 left

arulsuju@gmail.com Help End

Total Marks: 187.0

-hoice Questions

...hich of the following ... ts about Interface a... + 6.0

...u are transforming an ER ...del to a relational model, t... + 6.0

...Spring Boot, which of the ...owing codes is used to disa... + 4.0

...hich of these statements ...out Bean Factory in Spring is tr... + 6.0

...hich of these best completes the ...ne A in the following RE... + 6.0

...What is the return statement in the ...following Java code: ... + 4.0

...What is the output of the ...following RESTful web service + 4.0

Question 5

Max. Marks 6.00

Which of these best completes the Line A in the following REST code:

```
.......
HttpClient httpclient = new DefaultHttpClient();
                HttpPost httppost = new HttpPost(....);

                try {
                    List<NameValuePair> valuePairs = new ArrayList<NameValuePair>();
                      valuePairs.add(.....);
                      valuePairs.add(.....);

                      httppost.(_________)(new UrlEncodedFormEntity(valuePairs));  /*Line A*/
                      httpclient.execute(httppost);
              ......
}
```

setPath

setCollections

setParams

velopment

01:05:50 left

arulsuju@gmail.com Help

ks: 187.0

```
HttpClient httpclient = new DefaultHttpClient();
                HttpPost httppost = new HttpPost(....);

                try {
                    List<NameValuePair> valuePairs = new ArrayList<NameValuePair>();
                        valuePairs.add(.....);
                        valuePairs.add(.....);

                        httppost.(_________)(new UrlEncodedFormEntity(valuePairs));  /*Line A*/
                        httpclient.execute(httppost);
                ......
}
```

setPath

setCollections

setParams

setEntity

Question 7 Max. Marks 4.00

What is the output of the following RESTful web service code assuming that it includes a working **UserService** class?

```
 ....
class Web{
private String REST_SERVICE_URL = "http://localhost:8080/UserManagement/rest/UserService/users";
 private static final String SUCCESS_RESULT="<result>success</result>";
    .....
private void adduser(){
      Form form = new Form();
      form.param("id", "2");
      form.param("name", "suresh");
      form.param("profession", "CFO");
      String callResult = client
         .target(REST_SERVICE_URL)
         .request(MediaType.APPLICATION_XML)
         .post(Entity.entity(form,
            MediaType.APPLICATION_FORM_URLENCODED_TYPE),
            String.class);
       String result = "Pass";
       if(!SUCCESS_RESULT.equals(callResult)){
          result = "fail";
       }

       System.out.println("Test case name: testAddUser, Result: " + result );
```

```
                .build();

    Response response = client.newCall(request).execute();

     return (......);

}
```

- Response.return()
- response.tostring()
- Response.get()
- response.body().string()

Submit

Question 8

Max. Marks 6

Which of these is used to initialize the **WebResource** class object in the following RESTful API code:

```
....
 private static final String BASE_URI = "http://localhost:8083/Restful-web-service";
 private static final String PATH_NAME = "/PostExampleService/empInfo";
private static void post
{
ClientConfig config = new DefaultClientConfig();
        Client client = Client.create(config);
        WebResource resource =(....);
        WebResource nameResource = resource.path("rest").path(PATH_NAME);
......
}
```

WebResource.resource(Base_URI)

client.resource(PATH_NAME)

client.resource(BASE_URI)

client.path(BASE_URI)

Question 8

Max. Mar

Which of these is used to initialize the **WebResource** class object in the following RESTful API code:

```
....
 private static final String BASE_URI = "http://localhost:8083/Restful-web-service";
 private static final String PATH_NAME = "/PostExampleService/empInfo";
private static void post
{
ClientConfig config = new DefaultClientConfig();
        Client client = Client.create(config);
        WebResource resource =(....);
        WebResource nameResource = resource.path("rest").path(PATH_NAME);
 ......
 }
```

WebResource.resource(Base_URI)

client.resource(PATH_NAME)

client.resource(BASE_URI)

client.path(BASE_URI)

Question 12 Max. Marks 25.00

Maximum runs

Write a query to find the maximum runs scored by a batsman in the month of January 2016 among all the batsmen who played during that month.

Sample input

Table: CricketMatch

Field	Type
BatsmanName	Text
RunsMade	Integer
MatchDate	Text

Sample

BatsmanName	RunsMade	MatchDate
Colin	100	1-Jan-2016
Colin	50	5-Jan-2016
Colin	150	25-Jan-2016
Mike	120	20-Jan-2016
Mike	100	28-Jan-2016

BatsmanName	Text
RunsMade	Integer
MatchDate	Text

Sample

BatsmanName	RunsMade	MatchDate
Colin	100	1-Jan-2016
Colin	50	5-Jan-2016
Colin	150	25-Jan-2016
Mike	120	20-Jan-2016
Mike	100	28-Jan-2016
Chris	50	28-Jan-2016
Chris	75	02-Jan-2016
Chris	100	05-Jan-2016
Chris	100	05-Feb-2016

Sample output

MAX_RUNS
300

Mike	100	28-Jan-2016
Chris	50	28-Jan-2016
Chris	75	02-Jan-2016
Chris	100	05-Jan-2016
Chris	100	05-Feb-2016

Sample output

MAX_RUNS
300

New Submission

MySQL 5.6

Question 9

Max. Marks 50.00

Forgotten Report

There are *N* cities in a state. You start your ride from the **first** city.
You have to visit all other cities exactly once and finally return to your origin city. After visiting each city, you collect the analysis report.
But when you reached the last unvisited city, you remembered that you did not collect the report from city *K*. So, now you decide to first collect the report from city *K* and then return to your home city.
Given the distances between each pair of cities, you are required to find the shortest possible distance of your whole journey.

INPUT

The input begins with *T* (Number of test cases).
Second line contains *K* (City No. Where you forgot to collect the report).
Third line contains *N* (Number of cities).
Next there are N lines, I^{th} line have exactly *N* numbers denoting *distance* from city *I* to all *N* cities.

OUTPUT

For each test case, print the Minimum Distance of total journey.
Answer for each test case should come in a new line.

CONSTRAINTS

$1 \leq T \leq 10$
$1 \leq N \leq 18$
$1 \leq K \leq N$

$1 < K < N$

$0 \leq dist(i, j) \leq 100$

Sample Input

```
1
2
4
0 1 15 6
2 0 7 3
9 6 0 12
10 4 8 0
```

Sample Output

```
20
```

Explanation

Path = $1 -> 2 -> 4 -> 3 -> 2 -> 1$

Note : He forgets the report at city 2, so in order to collect the report, he needs to visit some of the cities again.

Note: Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the co
will be run on multiple test cases. Therefore, your code must solve th s problem statement.

Time Limit: 1.0 second(s) for each input file.

Memory Limit: 256 MB

Source Limit: 1024 KB

Marking Scheme: Marks are awarded if any test case passes.

Allowed Languages: Java, Python

New Submission

All Submissions

6.0

+ 4.0

+ 4.0

+ 6.0

+ 50.0

+ 50.0

+ 20.0

Question 10 Max. Marks 50.00

Special Intervals

You are given *n* intervals which are termed as special intervals. Each interval is of a different type. Again, you are given a set of *q* non-special intervals.
For each non-special interval in the given q intervals, you have to find the number of different types of special intervals in that non-special interval.

Note: A special interval is inside a non-special interval if there exists a point *x* which belongs to both special interval and non-special interval.

Input format

- First line: *n* denoting the number of special intervals
- Next *n* lines: Two integers denoting *lspecial[i]* and *rspecial[i]* denoting the range *[l,r]* for the i^{th} special interval.
- Next line: *q* denoting the number of non-special intervals
- Next *q* lines: Two integers denoting *lnonspecial[i]* and *rnonspecial[i]* denoting the range *[l,r]* for the i^{th} non-special interval

Output format

Print *q* space-separated integers denoting the answer for each of the *q* non-special intervals.

Input Constraints

$1 \leq n \leq 10^5$

$10^9 \geq lspecial[i] \leq 10^9$

- Next line: *q* denoting the number of non-special intervals
- Next *q* lines: Two integers denoting *lnonspecial[i]* and *rnonspecial[i]* denoting the range *[l,r]* for special interval

Output format

Print *q* space-separated integers denoting the answer for each of the *q* non-special intervals.

Input Constraints

$1 \leq n \leq 10^5$

$-10^9 \leq lspecial[i] \leq 10^9$

$-10^9 \leq rspecial[i] \leq 10^9$

$1 \leq q \leq 5 * 10^4$

$-10^9 \leq lnonspecial[i] \leq 10^9$

$-10^9 \leq rnonspecial[i] \leq 10^9$

Sample Input

```
3
1 2
1 5
1 7
3
1 3
3 3
```

Sample Output

```
3 2 1
```

Input Constraints

$1 \leq n \leq 10^5$

$-10^9 \leq lspecial[i] \leq 10^9$

$-10^9 \leq rspecial[i] \leq 10^9$

$1 \leq q \leq 5 * 10^4$

$-10^9 \leq lnonspecial[i] \leq 10^9$

$-10^9 \leq rnonspecial[i] \leq 10^9$

Sample Input

```
3
1 2
1 5
1 7
3
1 3
3 3
6 7
```

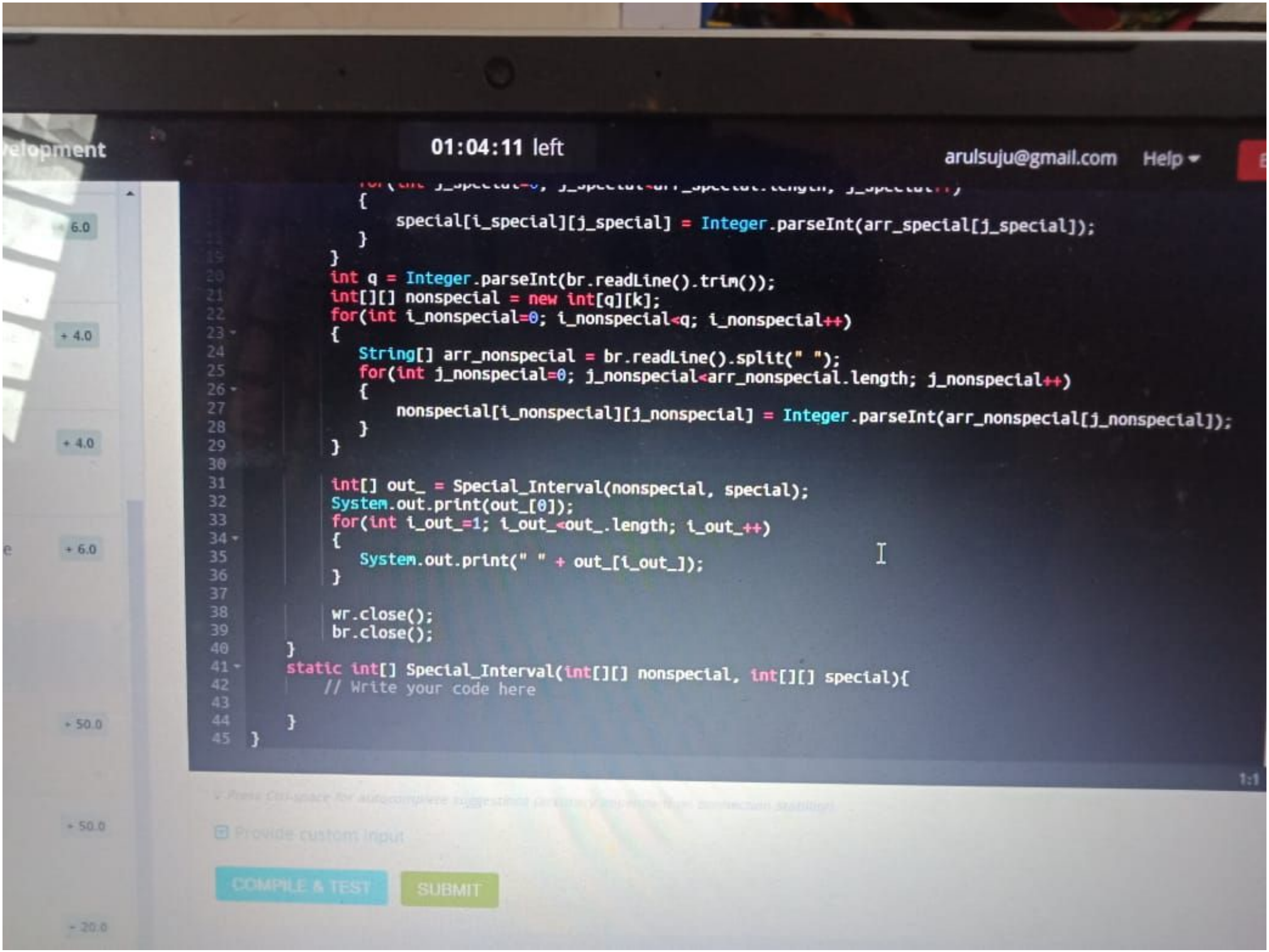
Sample Output

```
3 2 1
```

Explanation

For 1st non special interval [1,3] , there is atleast a point in each of the 3 special interval having a point common with the interval[1,3].Hence answer is 3.

Note: Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the will be run on multiple test cases. Therefore, your code must solve the problem statement.


01:04:11 left
arulsuju@gmail.com
Help
{
special[i_special][j_special] = Integer.parseInt(arr_special[j_special]);
}
}
int q = Integer.parseInt(br.readLine().trim());
int[][] nonspecial = new int[q][k];
for(int i_nonspecial=0; i_nonspecial<q; i_nonspecial++)
{
String[] arr_nonspecial = br.readLine().split(" ");
for(int j_nonspecial=0; j_nonspecial<arr_nonspecial.length; j_nonspecial++)
{
nonspecial[i_nonspecial][j_nonspecial] = Integer.parseInt(arr_nonspecial[j_nonspecial]);
}
}
int[] out_ = Special_Interval(nonspecial, special);
System.out.print(out_[0]);
for(int i_out_=1; i_out_<out_.length; i_out_++)
{
System.out.print(" " + out_[i_out_]);
}
wr.close();
br.close();
}
static int[] Special_Interval(int[][] nonspecial, int[][] special){
// Write your code here
}
}
Provide custom input
COMPILE & TEST
SUBMIT

+ 6.0

+ 4.0

+ 4.0

+ 6.0

+ 50.0

+ 50.0

+ 20.0

Question 11

Max. Marks 20.00

Convex hull problem

You are given a set S containing N points.

Write a program to find the number of points in the smallest subset of S that has the same convex hull as that of S.

Note:

- All the points in the input are not collinear.
- No two points are the same.

Input format

- First line: N
- Next N lines: Two space-separated integers x and y denoting the coordinates of S_i

Output format

Print the number of points in the smallest subset of S that has the same convex hull as that of S.

Constraints

$3 \leq N \leq 10^6$
$1 \leq x, y \leq 10^6$

Sample Input

5
1 1

Sample Output

4

$1 \leq x, y \leq 10^6$

Sample Input

```
5
1 1
1 5
3 3
5 5
5 1
```

Sample Output

```
4
```

Explanation

Convex hull is formed with points $\{1,1\}, \{1,5\}, \{5,5\}, \{5,1\}$. Thus answer is 4.

Note: Your code should be able to convert the sample input into the sample output. However, this is not enough to pass the challenge, because the will be run on multiple test cases. Therefore, your code must solve this problem statement.

Time Limit: 1.0 sec(s) for each input file.
Memory Limit: 256 MB
Source Limit: 1024 KB
Marking Scheme: Marks are awarded if any case is passed.
Allowed Languages: Java, Java 8

New Submission | All Submissions

Java (openjdk 1.7.0_95)

```
import java.io.*;
import java.util.*;


public class TestClass {
```

Java (openjdk 1.7.0_95)

Save

```java
import java.io.*;
import java.util.*;


public class TestClass {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        PrintWriter wr = new PrintWriter(System.out);
        int N = Integer.parseInt(br.readLine().trim());
        int[] X = new int[N];
        int[] Y = new int[N];
        for(int i=0; i<N; i++)
        {
            String[] inp = br.readLine().split(" ");
            X[i] = Integer.parseInt(inp[0]);
            Y[i] = Integer.parseInt(inp[1]);
         }
        int out_ = solve(N, X, Y);
        System.out.println(out_);

        wr.close();
        br.close();
    }
    static int solve(int N, int[] X, int[] Y){
        // Your code goes here
    }
}
```

Provide custom input

COMPILE & TEST

SUBMIT