```
    }
}
int main()
{
    int number,x;

    number = 5671;

    x=fun(number);

    printf("%d",x);

    return 0;
}
```

- 4
- 19
- 14
- 1765

Reset Answer

‹ Previous Question
Next Question ›

Question 5

Max. score: 6.00

What is the output of the following C code:

Code:

```
#include<stdio.h>
int main()
{
    int i, n;
    char *x="Hacker";
    n = strlen(x);
    *x = x[n];
    for(i=0; i<=n; i++)
    {
        printf("%s ", x);
        x++;
    }
    printf("\n", x);
    return 0;
}
```

- Compilation error
- Run-time error
- er

1 2 3 4 5 6 7 8 9 10 11 12

```
{
    int i, n;
    char *x="Hacker";
    n = strlen(x);
    *x = x[n];
    for(i=0; i<=n; i++)
    {
        printf("%s ", x);
        x++;
    }
    printf("\n", x);
    return 0;
}
```

- Compilation error
- Run-time error
- er
- Hacker

Reset Answer

1 2 3 4 5 6 7 8 9 10 11 12

‹ Previous Question

Next Question ›

Question 6

Max. score: 2.00

In C++, which of the following is an example of the **static** polymorphism?

- Operator overloading
- Function overloading
- Both of these
- None of these

Reset Answer

‹ Previous Question
Next Question ›

1
2
3
4
5
6
7
8
9
10
11
12

1
2
3
4
5
6
7
8
9
10
11
12

Question 6

Max. score: 2.00

In C++, which of the following is an example of the **static** polymorphism?

- Operator overloading
- Function overloading
- Both of these
- None of these

Reset Answer

‹ Previous Question
Next Question ›

Question 7

Max. score: 2.00

Consider you are working on RDBMS. So based on your experience, you have concluded few statements about the RDBMS.

Which of the following statements is correct?

Statements

1. RDBMS contains Normalization.
2. Distributed databases are not supported by RDBMS.

- only 1
- only 2
- Both 1 and 2
- None of these

Reset Answer

‹ Previous Question
Next Question ›

Code

```
<style>
    .cell,
    .tab {
        border: 0.3px solid black;
        border-collapse: collapse;
    }
</style>
<table class="tab">
    <tbody>
        <tr>
            <th class="cell">Sl No.</th>
            <th class="cell">Username</th>
            <th class="cell">Status 1 </th>
            <th class="cell">Status 2 </th>
        </tr>
        <tr>
            <td class="cell">1</td>
            <td class="cell">John</td>
            <td class="cell">Logged out</td>
            <td class="cell">Logged out</td>
        </tr>

        <tr>
            <td class="cell">2</td>
            <td class="cell">Ben</td>
            <td class="cell" rowspan="2">Logged in</td>
            <td class="cell" rowspan="2">Logged in</td>
        </tr>
```

Question 8

Max. score: 6.00

What is the output of the following HTML snippet:

Code

```
<style>
    .cell,
    .tab {
        border: 0.3px solid black;
        border-collapse: collapse;
    }
</style>
<table class="tab">
    <tbody>
        <tr>
            <th class="cell">Sl No.</th>
            <th class="cell">Username</th>
            <th class="cell">Status 1 </th>
            <th class="cell">Status 2 </th>
        </tr>
        <tr>
            <td class="cell">1</td>
            <td class="cell">John</td>
            <td class="cell">Logged out</td>
            <td class="cell">Logged out</td>
        </tr>

        <tr>
```

```
<table class="tab">
    <tbody>
        <tr>
            <th class="cell">Sl No.</th>
            <th class="cell">Username</th>
            <th class="cell">Status 1 </th>
            <th class="cell">Status 2 </th>
        </tr>
        <tr>
            <td class="cell">1</td>
            <td class="cell">John</td>
            <td class="cell">Logged out</td>
            <td class="cell">Logged out</td>
        </tr>

        <tr>
            <td class="cell">2</td>
            <td class="cell">Ben</td>
            <td class="cell" rowspan="2">Logged in</td>
            <td class="cell" rowspan="2">Logged in</td>
        </tr>
        <tr>
            <td class="cell">3</td>
            <td class="cell">Anita</td>
            <td class="cell" rowspan="2">Logged in</td>

        </tr>
        <tr>
            <td class="cell">4</td>
            <td class="cell">Mary</td>
```

```
<tr>
    <td class="cell">1</td>
    <td class="cell">John</td>
    <td class="cell">Logged out</td>
    <td class="cell">Logged out</td>
</tr>

<tr>
    <td class="cell">2</td>
    <td class="cell">Ben</td>
    <td class="cell" rowspan="2">Logged in</td>
    <td class="cell" rowspan="2">Logged in</td>
</tr>
<tr>
    <td class="cell">3</td>
    <td class="cell">Anita</td>
    <td class="cell" rowspan="2">Logged in</td>

</tr>
<tr>
    <td class="cell">4</td>
    <td class="cell">Mary</td>
    <td class="cell">Logged out</td>

</tr>
<tr>
    <td class="cell">5</td>
    <td class="cell">Mark</td>
    <td class="cell">Logged in</td>
    <td class="cell">Logged in</td>
```

```
            <td class="cell">2</td>
            <td class="cell">Ben</td>
            <td class="cell" rowspan="2">Logged in</td>
            <td class="cell" rowspan="2">Logged in</td>
        </tr>
        <tr>
            <td class="cell">3</td>
            <td class="cell">Anita</td>
            <td class="cell" rowspan="2">Logged in</td>

        </tr>
        <tr>
            <td class="cell">4</td>
            <td class="cell">Mary</td>
            <td class="cell">Logged out</td>

        </tr>
        <tr>
            <td class="cell">5</td>
            <td class="cell">Mark</td>
            <td class="cell">Logged in</td>
            <td class="cell">Logged in</td>
        </tr>

    </tbody>
</table>
```

Options

1

```
</table>
```

Options

1

Sl No.	Username	Status 1	Status 2
1	John	Logged out	Logged in
2	Ben	Logged in	
3	Anita		Logged in
4	Mary	Logged out	
5	Mark	Logged in	Logged in

2

Sl No.	Username	Status 1	Status 2	Status 3
1	John	Logged out	Logged out	
2	Ben	Logged in	Logged in	
3	Anita			Logged in
4	Mary	Logged out		
5	Mark	Logged in	Logged in	

3

Sl No.	Username	Status 1	Status 2	
1	John	Logged out	Logged out	
2	Ben	Logged in	Logged in	
3	Anita			Logged in
4	Mary	Logged out		
5	Mark	Logged in	Logged in	

4

3

Sl No.	Username	Status 1	Status 2	
1	John	Logged out	Logged out	
2	Ben	Logged in	Logged in	
3	Anita			Logged in
4	Mary	Logged out		
5	Mark	Logged in	Logged in	

4

Sl No.	Username	Status
1	John	Logged out
4	Mary	
2	Ben	Logged in
3	Anita	
5	Mark	

- 1
- 2
- 3
- 4

Reset Answer

Question 9

Max. score: 2.00

In C++, which of the following are the types of polymorphism?

- Run-time polymorphism
- Compile time polymorphism
- Both of these
- None of these

Reset Answer

‹ Previous Question
Next Question ›

1
2
3
4
5
6
7
8
9
10
11
12

Question 9

Max. score: 2.00

In C++, which of the following are the types of polymorphism?

- Run-time polymorphism
- Compile time polymorphism
- Both of these — Your answer has been saved.
- None of these

Reset Answer

‹ Previous Question
Next Question ›

1 2 3 4 5 6 7 8 9 10 11 12

Question 10

Max. score: 6.00

In Data structures, what is the best case and worst case complexity analysis for the following pseudocode in the insertion sort algorithm:

```
final Node<E> sorted = null;
    ........
    Node<E> current = first;
    while (current != NULL)
    {

        Node<E> next = current.next;
         Unlink(current);
        sortedInsert(current.data);
        current = next;
      .......
     .........
    }
```

- O(log n), O(n)
- O(n^2), O(n^2)
- O(n), O(n^2)
- None of these

Question 10

Max. score: 6.00

In Data structures, what is the best case and worst case complexity analysis for the following pseudocode in the insertion sort algorithm:

```
final Node<E> sorted = null;
    ........
    Node<E> current = first;
    while (current != NULL)
    {

        Node<E> next = current.next;
         Unlink(current);
        sortedInsert(current.data);
        current = next;
      .......
     .........
    }
```

- O(log n), O(n)
- O(n^2), O(n^2)
- O(n), O(n^2)
- None of these

```
Node<E> current = first;
while (current != NULL)
{

    Node<E> next = current.next;
     Unlink(current);
    sortedInsert(current.data);
    current = next;
  .......
 .........
}
```

- O(log n), O(n)
- O(n^2), O(n^2)
- O(n), O(n^2)
- None of these

Reset Answer

‹ Previous Question
Next Question ›

Question 12 Max. score: 20.00

Maximum balls

Bob is a maths teacher of a class of *N* students. The examination results were out and marks obtained by the students are represented by array *marks*. Bob is happy with the performance of the class so he decided to gift every student in his class an equal number of balls. As Bob likes mathematics a lot, he decided to pick a number which is a factor of *marks* of all *N* students in his class and decided to gift that number of balls to every student.

As Bob wants to distribute the maximum number of balls, he can change the marks of at most **1** student in his class to any positive number of his choice, so that he can gift the maximum number of balls to the students.

Task

Determine the maximum number of balls Bob can give to every student after changing the marks of at most *1* student to any positive number of his choice.

Example

Assumptions

- *N* = *3*
- *marks* = $[12, 3, 11]$

Approach

- The maximum possible number of balls each student can have is

New Submission All Submissions

Auto-complete connection failed. Reconnect Save Python 3 (python 3.9.5)

```
def max_balls (N, marks):
    # Write your code here


N = int(input())
marks = list(map(int, input().split()))

out_ = max_balls(N, marks)
print (out_)
```

3:5 vscode

Test against custom input
Custom input populated

Compile & Test code Submit code

< Previous Question Next Question >

As Bob wants to distribute the maximum number of balls, he can change the marks of at most **1** student in his class to any positive number of his choice, so that he can gift the maximum number of balls to the students.

Task

Determine the maximum number of balls Bob can give to every student after changing the marks of at most *1* student to any positive number of his choice.

Example

Assumptions

- *N = 3*
- *marks* = $[12, 3, 11]$

Approach

- The maximum possible number of balls each student can have is *3*.
- Here to distribute the maximum number of balls to every student, Bob can change the marks of the 3^{rd} student to *39*.
- So the answer to this sample example becomes *3*.

Function description

Complete the *max_balls* function provided in the editor. This function takes the following *2* parameters and returns an integer that represents the answer to the task as described in the problem statement:

- *N*: Represents the number of students in the class

New Submission | All Submissions

Auto-complete connection failed. Reconnect | Save | Python 3 (python 3.9.5)

```
def max_balls (N, marks):
    # Write your code here

N = int(input())
marks = list(map(int, input().split()))

out_ = max_balls(N, marks)
print (out_)
```

3:5 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >

Example

Assumptions

- $N = 3$
- $marks = [12, 3, 11]$

Approach

- The maximum possible number of balls each student can have is *3*.
- Here to distribute the maximum number of balls to every student, Bob can change the marks of the 3^{rd} student to *39*.
- So the answer to this sample example becomes *3*.

Function description

Complete the *max_balls* function provided in the editor. This function takes the following *2* parameters and returns an integer that represents the answer to the task as described in the problem statement.

- *N*: Represents the number of students in the class
- *marks:* Represents the marks of *N* students

Input format

Note: This is the input format that you must use to provide custom input (available above the **Compile and Test** button).

- The first line contains the function parameter *N*.
- The second line contains *N* space-separated integers denoting the input values of the function parameter *marks*.

New Submission | All Submissions

Auto-complete connection failed. Reconnect | Save | Python 3 (python 3.9.5)

```
def max_balls (N, marks):
    # Write your code here

N = int(input())
marks = list(map(int, input().split()))

out_ = max_balls(N, marks)
print (out_)
```

3:5 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >

Input format

Note: This is the input format that you must use to provide custom input (available above the **Compile and Test** button).

- The first line contains the function parameter *N*.
- The second line contains *N* space-separated integers denoting the input values of the function parameter *marks*.

Output format

Print an integer representing the maximum number for the given task.

Constraints

$2 \leq N \leq 10^5$

$1 \leq marks_i \leq 10^9$

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input

```
3
12 3 11
```

Sample output

```
3
```

New Submission | All Submissions

Auto-complete connection failed. Reconnect | Save | Python 3 (python 3.9.5)

```python
def max_balls (N, marks):
    # Write your code here


N = int(input())
marks = list(map(int, input().split()))

out_ = max_balls(N, marks)
print (out_)
```

3:5 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >

Explanation

For the given sample input

Given

- $N = 3$
- $marks = [12, 3, 11]$

Approach

- The maximum possible number of balls each student can have is *3*.
- Here to distribute the maximum number of balls to every student, Bob can change the marks of the 3^{rd} student to *39*.
- So the answer to this given input becomes *3*.

The following test cases are the actual test cases of this question that may be used to evaluate your submission.

Note:

Your code must be able to print the sample output from the provided sample input. However, your code is run against multiple hidden test cases. Therefore, your code must pass these hidden test cases to solve the problem statement.

Limits

Time Limit: 10 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB

Scoring

Score is assigned if any testcase passes

Allowed Languages

Bash, C, C++, C++14, C++17, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, Java 14, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Python 3.8, Racket, Ruby, Rust, Scala, Swift-4.1, Swift, TypeScript, Visual Basic

New Submission | All Submissions

Auto-complete connection failed. Reconnect | Save | Python 3 (python 3.9.5)

```
1  def max_balls (N, marks):
2      # Write your code here
3
4
5  N = int(input())
6  marks = list(map(int, input().split()))
7
8  out_ = max_balls(N, marks)
9  print (out_)
```

3:5 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

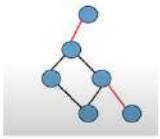
< Previous Question | Next Question >

Question 11 Max. score: 50.00

Special edge

You are given a connected undirected graph having *N* nodes numbered from *1* to *N* and *M* edges between its nodes. It is guaranteed that the input graph is connected and consists of no self-loops and no multiple edges between two vertices. Edges are given in a 2D array/vector *Edg*.

An edge is *special* if it is removed then the number of connected components in the graph increases.



Edges marked red are special in the above figure.

Task

Determine the number of unordered pairs of nodes *(u,v)* such that each and every Simple path (path with no edges repeated) between node *u* and node *v* consists of *exactly 1* special edge.

Notes:

- Assume 1-based indexing.
- A *connected undirected graph* is a network of nodes connected by bidirectional edges such that we can move between any two

New Submission All Submissions

Auto-complete connection failed. Reconnect Save Python 3 (python 3.9.5)

```python
def special_sol (N, M, Edg):
    """
    N - number of vertices in the graph
    M - number of edges
    There is an
    edge between Edg[i][0] and Edg[i][1]
    """

    pass

T = int(input())
for _ in range(T):
    N = int(input())
    M = int(input())
    Edg = [list(map(int, input().split())) for i in range(M)]

    out_ = special_sol(N, M, Edg)
    print (out_)
```

7:6 vscode

Test against custom input
Custom input populated

Compile & Test code Submit code

< Previous Question Next Question >

Task

Determine the number of unordered pairs of nodes *(u,v)* such that each and every Simple path (path with no edges repeated) between node *u* and node *v* consists of *exactly 1* special edge.

Notes:

- Assume 1-based Indexing.
- A *connected undirected graph* is a network of nodes connected by bidirectional edges such that we can move between any two nodes using some bidirectional edges.
- A simple path between 2 vertices contains distinct vertices only.
- Unordered pair means that *(a, b)* and *(b, a)* are considered to be the same.

Example

Assumptions

- *N* = 4
- *M* = 4
- *Edg = [[1,2], [2,3], [3,4], [2,4]]*

Approach

- The paths between the nodes *(1, 2), (1, 3), (1, 4)* consist of exactly one Special edge which is the edge *[1, 2]*.
- Hence, the answer is *3*.

Function description

Complete the function *special_sol* provided in the editor. This function takes the following *3* parameters and returns the required answer:

New Submission | All Submissions

Auto-complete connection failed. Reconnect | Save | Python 3 (python 3.9.5)

```
def special_sol (N, M, Edg):
    """
    N - number of vertices in the graph
    M - number of edges
    There is an
    edge between Edg[i][0] and Edg[i][1]
    """

    pass

T = int(input())
for _ in range(T):
    N = int(input())
    M = int(input())
    Edg = [list(map(int, input().split())) for i in range(M)]

    out_ = special_sol(N, M, Edg)
    print (out_)
```

7:6 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >

Example

Assumptions

- *N = 4*
- *M = 4*
- *Edg = [[1,2], [2,3], [3,4], [2,4]]*

Approach

- The paths between the nodes *(1, 2), (1, 3), (1, 4)* consist of exactly one Special edge which is the edge *[1, 2]*.
- Hence, the answer is *3*.

Function description

Complete the function *special_sol* provided in the editor. This function takes the following *3* parameters and returns the required answer:

- *N*: Represents the number of nodes in the graph
- *M*: Represents the number of edges in the graph
- *Edg*: Represents a list of edges between two nodes in the graph

Input format

- The first line contains *T* denoting the number of test cases. *T* also specifies the number of times you have to run the *special_sol* function on a different set of inputs.
- For each test case:
  - The first line contains a single integer *N* denoting the number of nodes in the graph
  - The second line contains a single integer *M* denoting the

New Submission | All Submissions

Auto-complete ready! Save Python 3 (python 3.9.5)

```
def special_sol (N, M, Edg):
    """
    N - number of vertices in the graph
    M - number of edges
    There is an
    edge between Edg[i][0] and Edg[i][1]
    """

    pass

T = int(input())
for _ in range(T):
    N = int(input())
    M = int(input())
    Edg = [list(map(int, input().split())) for i in range(M)]

    out_ = special_sol(N, M, Edg)
    print (out_)
```

7:6 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >

Input format

- The first line contains *T* denoting the number of test cases. *T* also specifies the number of times you have to run the *special_sol* function on a different set of inputs.
- For each test case:
  - The first line contains a single integer *N* denoting the number of nodes in the graph
  - The second line contains a single integer *M* denoting the number of edges in the graph
  - Next M lines contain integers u_i and v_i denoting that there is an edge between vertex u_i and vertex v_i

Output format

For each test case in a new line, print an integer denoting the number of unordered pairs *(u, v)* such that each and every simple path between *u* and *v* consists of exactly *1* special edge.

Constraints

$1 \leq T \leq 10$

$1 \leq N, M \leq 2 \times 10^5$

$1 \leq u_i, v_i \leq N$

$u_i \neq v_i$

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input

Sample output

New Submission | All Submissions

Auto-complete ready! | Save | Python 3 (python 3.9.5)

```python
def special_sol (N, M, Edg):
    """
    N - number of vertices in the graph
    M - number of edges
    There is an
    edge between Edg[i][0] and Edg[i][1]
    """

    pass

T = int(input())
for _ in range(T):
    N = int(input())
    M = int(input())
    Edg = [list(map(int, input().split())) for i in range(M)]

    out_ = special_sol(N, M, Edg)
    print (out_)
```

7:6 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >

Explanation

The first line contains the number of test cases, $T = 1$.

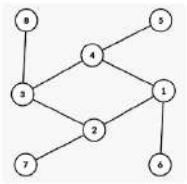
The first test case

Given

- $N = 8$
- $M = 8$
- $Edg = [[1, 6], [2, 7], [3, 8], [4, 5], [1, 2], [2, 3], [3, 4], [4, 1]]$

Approach

- The graph is as follows:



- The Edges $[1, 6], [2, 7], [3, 8], [4, 5]$ are Special.
- The following are ordered pairs of nodes having exactly one special edge between each and every simple path between them: $(6, 1), (6, 2), (6, 3), (6, 4), (7, 1), (7, 2), (7, 3), (7, 4), (8, 1), (8, 2), (8, 3), (8, 4), (5, 1), (5, 2), (5, 3), (5, 4)$.

Hence, the answer is 16.

The following test cases are the actual test cases of this question that may be used to evaluate your submission.

Note:

Your code must be able to print the sample output from the provided sample input. However, your code is run against multiple hidden test cases. Therefore

New Submission | All Submissions

Auto-complete ready! | Save | Python 3 (python 3.9.5)

```python
def special_sol (N, M, Edg):
    """
    N - number of vertices in the graph
    M - number of edges
    There is an
    edge between Edg[i][0] and Edg[i][1]
    """

    pass

T = int(input())
for _ in range(T):
    N = int(input())
    M = int(input())
    Edg = [list(map(int, input().split())) for i in range(M)]

    out_ = special_sol(N, M, Edg)
    print (out_)
```

7:6 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >

Code snippets (also called starter code/boilerplate code)

This question has code snippets for C, CPP, Java, and Python.

Sample input

```
1
8
8
1 6
2 7
3 8
4 5
1 2
2 3
3 4
```

Sample output

```
16
```

View more

Explanation

The first line contains the number of test cases, $T = 1$.

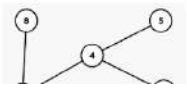
The first test case

Given

- $N = 8$
- $M = 8$
- $Edg = [[1, 6], [2, 7], [3, 8], [4, 5], [1, 2], [2, 3], [3, 4], [4, 1]]$

Approach

- The graph is as follows:



New Submission | All Submissions

Auto-complete ready! | Save | Python 3 (python 3.9.5)

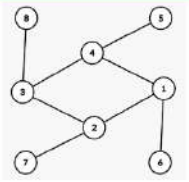
```python
def special_sol (N, M, Edg):
    """
    N - number of vertices in the graph
    M - number of edges
    There is an
    edge between Edg[i][0] and Edg[i][1]
    """

    pass

T = int(input())
for _ in range(T):
    N = int(input())
    M = int(input())
    Edg = [list(map(int, input().split())) for i in range(M)]

    out_ = special_sol(N, M, Edg)
    print (out_)
```

7:6 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >



- The Edges *[1, 6], [2, 7], [3, 8], [4, 5]* are Special.
- The following are ordered pairs of nodes having exactly one special edge between each and every simple path between them: *(6, 1), (6, 2), (6, 3), (6, 4), (7, 1), (7, 2), (7, 3), (7, 4), (8, 1), (8, 2), (8, 3), (8, 4), (5, 1), (5, 2), (5, 3), (5, 4).*

Hence, the answer is *16*.

The following test cases are the actual test cases of this question that may be used to evaluate your submission.

Note:

Your code must be able to print the sample output from the provided sample input. However, your code is run against multiple hidden test cases. Therefore, your code must pass these hidden test cases to solve the problem statement.

Limits

Time Limit: 2.0 sec(s) for each input file
Memory Limit: 256 MB
Source Limit: 1024 KB

Scoring

Score is assigned if any testcase passes.

Allowed Languages

Bash, C, C++, C++14, C++17, Clojure, C#, D, Erlang, F#, Go, Groovy, Haskell, Java, Java 8, Java 14, JavaScript(Rhino), JavaScript(Node.js), Julia, Kotlin, Lisp, Lisp (SBCL), Lua, Objective-C, OCaml, Octave, Pascal, Perl, PHP, Python, Python 3, Python 3.8, Racket, Ruby, Rust, Scala, Swift-4.1, Swift, TypeScript, Visual Basic

New Submission | All Submissions

Auto-complete ready! | Save | Python 3 (python 3.9.5)

```python
def special_sol (N, M, Edg):
    """
    N - number of vertices in the graph
    M - number of edges
    There is an
    edge between Edg[i][0] and Edg[i][1]
    """

    pass

T = int(input())
for _ in range(T):
    N = int(input())
    M = int(input())
    Edg = [list(map(int, input().split())) for i in range(M)]

    out_ = special_sol(N, M, Edg)
    print (out_)
```

7:6 vscode

Test against custom input
Custom input populated

Compile & Test code | Submit code

< Previous Question | Next Question >