Tata Elxsi - Final Practice

Test Summary

- No. of Sections: 2
- No. of Questions: 44
- Total Duration: 195 min

Section 1 - MCQ

Section Summary

- No. of Questions: 40
- Duration: 45 min

Additional Instructions:

None

Q1. How is named the bean that is defined in the following configuration class. Select a single answer.

```
@Configuration
public class ApplicationConfig
{
   @Autowired
   private DataSource dataSource;
   @Bean
   ClientRepository clientRepository()
   {
      ClientRepository accountRepository = new JpaClientRepository();
      accountRepository.setDataSource(dataSource);
      return accountRepository;
   }
}


```

- JpaClientRepository
- clientRepository
- Two beans are defined : a data source and a repository
- None of the above

Q2. What is the output of the following application?

```
package finance;
enum Currency
{ DOLLAR, YEN, EURO }
abstract class Provider
{
   protected Currency c = Currency.EURO;
}
public class Bank extends Provider
{
   protected Currency c = Currency.DOLLAR;
   public static void main( String[] pennies)
   {
      int value = 0;
      switch( new Bank(). c)
      {
         case 0:value--; break;
         case 1: value + +; break;
      }
      System.out.print( value);
   }
}

```

- 0
- 1
- The code does not compile.
- The code compiles but throws an exception at runtime.

Q3. For the following function,

```
int example (int x, int y)
{
  int z = 0;
  if ((x>0) && (y>0))
  {
    z = x;
  }
  return z;
}
```

Which function call ensure statement coverage for this function?

- example(0,0)
- example(1,0)
- example(0,1)
- example(1,1)

Q4. Which best describes this code?

```
1  class Stats {
2     private int data;
3     public int getData() {
4        return data;
5     }
6     public void setData(int data) {
7        this.data = data;
8     }
9  }
10
11
```

- It is a singleton.
- It is well encapsulated.
- It is immutable.
- It is both well encapsulated and immutable.

Q5. What will be the output of the following Java program?
Command line execution is done as – “java Output This is a command Line”.

```
1  class Output
2     {
3        public static void main(String args[])
4        {
5           System.out.print("args[0]");
6        }
7     }
```

java

Output

This

is

Q6. What's true about the following @Entity association between Department and Employee?

```
@Entity
public class Employee {
@Id
private int empId;
@ManyToOne
private Department dept;
}
@Entity
public class Department {
@Id
private int deptId;
@OneToMany(mappedBy="dept")
private List<Employee> employees;
}
```

It's OneToMany unidirectional association

It's OneToMany bidirectional association

The association owner is the Department class

None of the above

Q7. What is the output of this program?

```
1  public interface A
2  {
3     protected String getName();
4  }
5  public class Test implements A
6  {
7     public String getName()
8     {
9        return "name";
10    }
11    public static void main (String[] args)
12    {
13       Test t = new Test();
14       System.out.println(t.getName());
15    }
16 }
17
```

name

compilation error due to protected method

compilation error in method definition

runtime exception

```
1  class Turbo extends Motor {
2     public void startup() {
```

```
    public void startup() {
        System.out.print("turbo-");
    }
}

public class Motor {
    public void startup() {
        System.out.print("motor-");
    }
    public static void main( String[] args) {
        Motor motor = new Turbo();
        Turbo turbo = new Turbo();

        motor.startup();
        turbo.startup();
    }
}
```

- motor-turbo-
- turbo-motor-
- turbo-turbo-
- None of the above

Q9. Which of these classes properly implement(s) the singleton pattern?

```
class Bugs {
    private static Bugs instance = new Bugs();
    private List < String > bugs = new ArrayList < >();
    public static List < String > getAnswers() {
        return instance.bugs;
    }
}

class Defects {
    private static Defects instance = new Defects();
    private List < String > problems = new ArrayList < >();
    public static Defects getDefects() {
        return instance;
    }
    public List < String > getProblems() {
        return problems;
    }
}
```

- Bugs
- Defects
- Both classes
- Neither class

Q10. Following Docker command:
docker exec -it container_id bash
is used to:

- Activate default VM machine
- Access a running container
- Both a and b

None of the above

Q11. The package for the assertThat() function is?

org.hamcrest.CoreMatchers.assertThat

org.junit.Assert.assertThat

org.junit.JunitMatchers.assertThat

org.junit.hasItem.assertThat

Q12. What will be the output?

```
public class Test{
    public static void main(String[] args){
        int[] a = new int[4];
        a[1] = 1;
        a = new int[2];
        System.out.println("a[1] is " + a[1]);
    }
}
```

A. The program has a compile error because new int[2

B. The program has a runtime error because a[1

C. a[1] is 0

D. a[1] is 1

Q13. John wants to run his Spring Boot Application on port 8585 of his local machine. Which property he needs to write to change the application port to 8585

application.port=8585

spring.port=8585

server.port=8585

app.default.port=8585

Q14. What is the output of the given snippet?

```
<div *ngIf="false; else elseBlock"> Server 1 Works</div>
<ng-template #elseBlock>Server 2 Works</ng-template>
```

Server 1 Works

Server 2 Works

- Both Server 1 and Serve 2 works
- Error

Q15. Endpoint Classes for XML marshalling

- AbstractDomPayloadEndpoint
- AbstractSaxPayloadEndpoint
- AbstractStaxStreamPayloadEndpoint
- AbstractMarshallingPayloadEndpoint

Q16. _____ is written in a programming language and is a short program used to test part of functionality of the software system.

- Test Scenarios
- Test Cases
- Test Script
- Latent defect

Q17. Endpoint Classes for DOM

- AbstractDomPayloadEndpoint
- AbstractJDomPayloadEndpoint
- AbstractDom4jPayloadEndpoint
- AbstractXomPayloadEndpoint

Q18. Consider the following code snippet

```
@RestController
public class EmployeeController {
@PostMapping("/employee")
public Employee addEmployee(Employee employee){
return employee;
}
}
```

What data will be returned by this Api if the client makes POST a request to /employee and sends a valid employee object?

- It will return a response with same employee object as the response body
- It will return a response with an empty body
- It will generate a compilation error as @RequestBody is missing

None of the above

Q19. What is the output if status=[1,2,3]

```
<li *ngIf="status">TRUE</li>
<li *ngIf="status">FALSE</li>
```

TRUE

FALSE

Q20. What will be the output of this program?

```
interface Walk
{
   public default int getSpeed()
   {
      return 5;
   }
}
interface Run
{
   public default int getSpeed()
   {
      return 10;
   }
}
 public class Animal implements Walk,Run
 {
   public static void main(String args [ ] )
   {
      Animal an = new Animal();
      System.out.println(an.getSpeed());
   }
 }
```

5

10

An exception is thrown at run-time.

Compilation fails due to an error at line 15.

Compilation fails due to multiple errors.

Q21. Hari has written the following code snippet to create a Rest Api that takes a username and returns the user details.

```
@Controller
public class UserController {
@GetMapping("/users/{userName}")
public User getUser(@PathVariable String userName){
Â Â Â Â Â  return getUserByName(userName);
Â  }
}
```

** getUserByName(String userName) is a method defined inside the UserController class that fetches a User from the database.

The code compiled and the application started successfully. However when he made a request to the Api to fetch the User details, it encountered an error with a status code of 404.
What changes Hari should make to his code to resolve the issue?

Hari should add @ResponseBody annotation on top of getUser(..) method

Hari should replace the @Controller annotation with @RestController

Either of the above are correct

None of the above

Q22. Predict the behavior of the following code.

```
interface IShape {
   void f1();
   void f2();
   void f3();
}
class Circle implements IShape {
   public void f1() {
   }
}

```

Compile time error

Run time error

The code is correct

Exception

Q23. Attribute directives change the appearance or behavior of an element and then, Which of the following directives change the DOM layout by adding and removing DOM elements.

structural

advanced

component

same

Q24. How to use in spring framework?

<ref> is used with bean id.

<ref> is used with string values.

All of the above

None of the above

Q25. What is the role of ApplicationContextAware in spring?

Makes a bean aware to the container.

Dependency injection is performed.

All of the above

None of the above

Q26.

```
1  What would be the result of attempting to compile and run the following code?
2  public class HelloWorld{
3      public static void main(String[] args){
4          double[] x = new double[]{1, 2, 3};
5          System.out.println("Value is " + x[1]);
6      }
7  }
```

The program has a compile error because the syntax new double[]{1, 2, 3} is wrong and it should be replaced by {1, 2, 3}.

The program has a compile error because the syntax new double[]{1, 2, 3} is wrong and it should be replaced by new double[3]{1, 2, 3};

The program has a compile error because the syntax new double[]{1, 2, 3} is wrong and it should be replaced by new double[]{1.0, 2.0, 3.0};

The program compiles and runs fine and the output

Q27. Following Docker command:
docker commit -m "My first update" container_ID user_name/repository_name
is used to:

Activate default VM machine

Access a running container

Build an image

Commit changes done in a Docker image

Q28. Can we create a custom directive in Angular?

TRUE

FALSE

Q29. The closeTo function is found under which package?

org.hamcrest.TypeSafeMatcher

org.hamcrest.BaseMatcher

org.hamcrest.number.IsCloseTo

org.hamcrest.number.CloseTo

Q30. @SpringBootApplication annotation is a combination of the annotations _____

- @Configuration
- @ComponentScan
- @EnableAutoConfiguration
- All of the above

Q31. Given the following Spring configuration file, what is the correct answer:

```
<bean class=”com.spring.service.MyServiceImpl”>
<property name=”repository” ref=”jpaDao”/>
</bean>
<bean class=”com.spring.repository.JpaDao”/>
```

- The first declared bean MyServiceImpl is missing an id must be named myService
- The second declared bean JpaDao is missing an id must be named jpaDao
- Answers 1 and 2 are both right
- Answers 1 and 2 are both wrong

Q32. Which of the following is correct about Test Runner in JUnit?

- Test runner is used for executing the test cases.
- @RunWith and @Suite annotation are used to run the test runner.
- Both of the above
- None of the above.

Q33. Spring-WS provides various abstract endpoint classes for you to process the request.

- org.springframework.ws.server.endpoint
- org.springframework.ws.server
- org.springframework.*
- none of the mentioned

Q34. Which is the correct statement to fetch the index value in *ngFor?

- ngFor="let item of 3];Let i = index"

ngFor="let item of [1,2,3];Let i = index"

*ngFor="let item of 3;Let i = index"

*ngFor="let item of [1,2,3];Let i = index"

Q35. Endpoint Classes for JDOM

AbstractDomPayloadEndpoint

AbstractJDomPayloadEndpoint

AbstractDom4jPayloadEndpoint

AbstractXomPayloadEndpoint

Q36. Endpoint Classes for XOM

AbstractDomPayloadEndpoint

AbstractJDomPayloadEndpoint

AbstractDom4jPayloadEndpoint

AbstractXomPayloadEndpoint

Q37. In which spring scope can any number of instances of bean be created?

Prototype scope

Request scope

Session scope

None of the above

Q38. Following Docker command:
docker push user_name/repository_name
is used to:

Activate default VM machine

Push changes done in an docker

image into Docker HubBuild an image

Commit changes done in a Docker image

Q39. Docker containers are based on open standard_____.

- Allowing containers to run on all major Linux distributions only
- Allowing containers to run on all Microsoft operating systems only
- Allowing containers to run on all major Linux distributions and Microsoft operating systems
- None of the above

Q40. Following Docker command:
eval $(docker-machine env default)
is used to:

- Activate default VM machine
- Access a running container
- Build an image
- Both a and b

Section 2 - Hands-on

Section Summary

- No. of Questions: 4
- Duration: 150 min

Additional Instructions:

None

Q1. Manoj is working as a ticket checker in sathyam cinemas. Due to over crowding, everyone are asked to form a queue. so people who came together got separated in the queue. People who came together are considered to be a single group. Always assume that the person of each group who is standing first in the queue holds the ticket. You have to check the ticket of each person and make sure all the members of his group enter the screen along with him.

For example,
Imagine **"abcaaubcc"** as a queue and each alphabet represents a person standing in queue. People of same group are represented using same alphabet.

The first person you will be checking is **"a".** You have to allow all the people of group "**a".** Then the list will be like this "**aaabcubcc"**

The next person you will be checking is **"b"** now the list will be updated as **"aaabbcucc".** After checking c list will be **"aaabbcccu".** Then after checking "**u**" the list will be same as "**aaabbcccu**"

Note: People of same group will be represented using same alphabet character. Also only small case letters will be used.

Input Format

A series of Alphabet characters representing the persons standing in a queue.

Output Format

The series of Alphabet characters representing the people in the order they went inside movie hall.

Sample Input	Sample Output
abcaaubcc	aaabbcccu

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q2. **Version Management System**

A version Managementsystem (VMS) is a repository of files, often the files for the source code of computer programs, with monitored access. Every change made to the source is tracked, along with who made the change, why they made it, and references to problems fixed, or enhancements introduced, by the change.

In this problem we will consider a simplified model of a development project. Let's suppose that there are N source files in the project. All the source files are distinct and numbered from 1 to N.

A VMS which is used for maintaining the project contains two sequences of source files. The first sequence contains M source files that are ignored by the VMS. If a source file is not in the first sequence, then it's considered to be unignored. The second sequence contains K source files that are tracked by the VMS. If a source file is not in the second sequence, then it's considered to be untracked. A source file can either be or not be in any of these two sequences. Your task is to calculate two values: the number of source files of the project, that are both tracked and ignored, and the number of source files of the project, that are both untracked and unignored.

Input Format

The first line of the input contains three integers N, M and K denoting the number of source files in the project, the number of ignored source files and the number of tracked source files. Assume that the maximum value for N as 50.

The second line contains M distinct integers denoting the sequence A of ignored source files. The sequence is strictly increasing.

The third line contains K distinct integers denoting the sequence B of tracked source files. The sequence is strictly increasing.

Output Format

Output a single line containing two integers: the number of the source files, that are both tracked and ignored, and the number of the source files, that are both untracked and unignored.

Sample Input

```
7 4 6
1 4 6 7
1 2 3 4 6 7
```

Sample Output

```
4 1
```

Sample Input

```
4 2 2
1 4
3 4
```

Sample Output

```
1 1
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q3. **Inverted Hollow Pyramid**

The much awaited event at the entertainment industry every year is the "Screen Awards". This year the event is going to be organized on December 25 to honour the Artists for their professional excellence in Cinema. The Organizers of the event, J&R Events, decided to design some attractive and LED Matrix panel boards for the show promotions all across the venue.

The Event organizers wanted to program the display boards with some specific pattern using alphabets and special characters. Help them write a program to design the pattern of an inverted hollow pyramid in the matrix panel, given the number of lines of the pattern.

Input Format

First line of the input is an integer that refers to the number of lines in the pattern.

Output Format

Output the pattern as given in the output.

Sample Input

```
4
```

Sample Output

```
*******
b*iii*b
bb*i*bb
bbb*bbb
```

Sample Input

```
5
```

Sample Output

```
*********
b*iiiii*b
bb*iii*bb
bbb*i*bbb
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Q4. **Valid Configuration**

Nurikabe logical game (sometimes called Islands in the Stream) is a binary determination puzzle. The puzzle is played on a typically rectangular grid of cells, some of which contain numbers. You must decide for each cell if it is white or black (by clicking on them) according to the following rules:

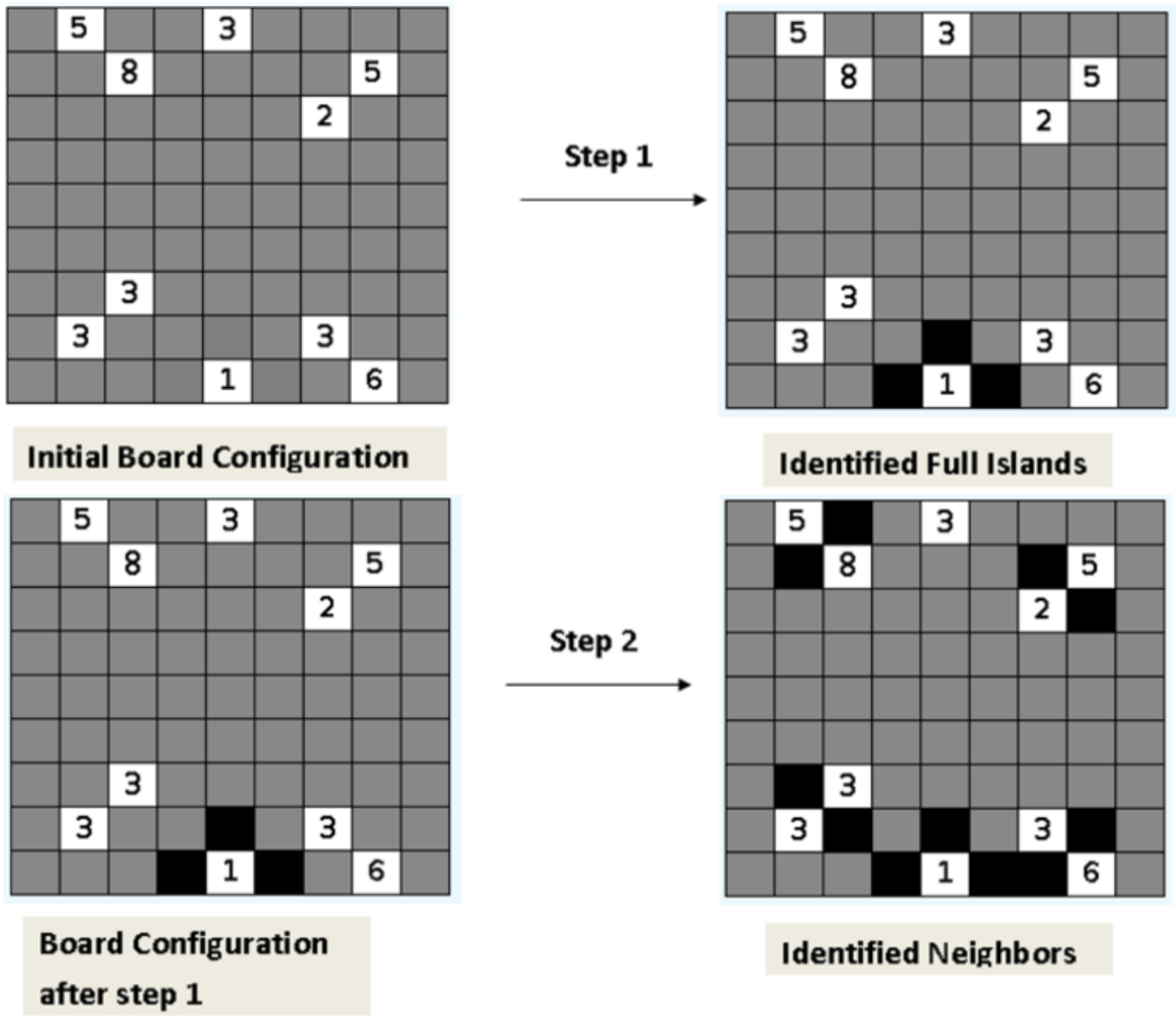
- All of the black cells must be connected.
- Each numbered cell must be part of a white island of connected white cells.
- Each island must have the same number of white cells as the number it contains (including the numbered cell).
- Two islands may not be connected.
- There cannot be any 2x2 blocks of black cells.

Unnumbered cells start out grey and cycle through white and black when clicked. Initially numbered cells are white in color.

Problem Statement:

The step 1 of solving the puzzle is identifying "Full islands".

An island is full if it contains as many white cells as the number in the region. Any 1s are trivially full regions. When you encounter a full region, any cells that boarder it must be black. Here we show the cells that must be black due to a single celled white island.
Below figure is the one after identifying full islands
The step 2 of solving the puzzle is to identify the neighbors.
Since two numbers in a nurikabe puzzle cannot be part of the same island, any cell that has two numbered neighbors must be black.
The two cases are when a cell is between two numbered cells, or (as in the image) when two numbered cells in the nurikabe are adjacent diagonally.



Given a board configuration in which empty white cells are represented by -1, black cells are represented by 0 and grey cells are represented by 20. Write a program to find whether it is a valid configuration assuming it to be obtained after performing step 1 and 2.

Input Format

First and only line of input is an integer N that gives the number of rows and columns of the grid.
Next N lines will have a board configuration with N*N cells assuming it to be obtained after performing step 1 and step 2. Assume that the maximum number in a cell can be 10. Grey colored cells are represented by 20, empty white cells are represented by -1 and black cells are represented by 0 in the matrix representation of the input configuration.

Output Format

Output should display "Yes" (without quotes) if the given configuration is a valid one obtained after performing step 1 and 2 of the nurikabe puzzle. Print "No" otherwise.
Refer sample input and output for formatting specifications.

Sample Input

```
5
20 0 1 0 2
20 20 0 20 -1
20 20 20 20 20
```

Sample Output

```
Yes
```

Sample Input

```
5
20 0 1 0 2
20 20 0 20 -1
20 20 20 20 20
```

Sample Output

```
No
```

Time Limit: - ms Memory Limit: - kb Code Size: - kb

Answer Key & Solution

Section 1 - MCQ

Q1 clientRepository

Solution

No Solution

Q2 The code does not compile.

Solution

No Solution

Q3 example(1,1)

Solution

No Solution

Q4 It is well encapsulated.

Solution

No Solution

Q5 This

Solution

No Solution

Q6

Solution

No Solution

Q7 compilation error due to protected method

Solution

No Solution

Q8 turbo-turbo-

Solution

No Solution

Q9 Neither class

Solution

No Solution

Q10 Access a running container

Solution

No Solution

Q11 org.junit.Assert.assertThat

Solution

No Solution

Q12 C. a[1] is 0

Solution

No Solution

Q13 server.port=8585

Solution

No Solution

Q14 Server 2 Works

Solution

No Solution

Q15 AbstractMarshallingPayloadEndpoint

Solution

No Solution

Q16 Test Script

Solution

No Solution

Q17 AbstractDomPayloadEndpoint

Solution

No Solution

Q18

It will return a response with an empty body

Solution

No Solution

Q19

TRUE

Solution

No Solution

Q20

Compilation fails due to an error at line 15.

Solution

No Solution

Q21

Either of the above are correct

Solution

No Solution

Q22

Compile time error

Solution

No Solution

Q23

structural

Solution

No Solution

Q24

<ref> is used with bean id.

Solution

No Solution

Q25

Makes a bean aware to the container.

Solution

No Solution

Q26

The program compiles and runs fine and the output

Solution

new double[]{1, 2, 3} is correct. This is the syntax I have not covered in this edition, but will be covered in the future edition. In this question,

double[] x = new double[]{1, 2, 3} is equivalent to double[] x = {1, 2, 3};

Q27

Commit changes done in a Docker image

Solution

No Solution

Q28

TRUE

Solution

No Solution

Q29

org.hamcrest.number.IsCloseTo

Solution

No Solution

Q30

All of the above

Solution

No Solution

Q31

The second declared bean JpaDao is missing an id must be named jpaDao

Solution

No Solution

Q32

Test runner is used for executing the test cases.

Solution

No Solution

Q33

org.springframework.ws.server.endpoint

Solution

No Solution

Q34

*ngFor="let item of [1,2,3];Let i = index"

Solution

No Solution

Q35 AbstractJDomPayloadEndpoint

Solution

No Solution

Q36 AbstractXomPayloadEndpoint

Solution

No Solution

Q37 Prototype scope

Solution

No Solution

Q38 Push changes done in an docker

Solution

No Solution

Q39 Allowing containers to run on all major Linux distributions and Microsoft operating systems

Solution

No Solution

Q40 Activate default VM machine

Solution

No Solution

Section 2 - Hands-on

Q1

Test Case

Input	Output
testcase	tteessca

Weightage - 10

Input	Output
sample	sample

Weightage - 10

Input

```
jjjjjjjjj
```

Output

```
jjjjjjjjj
```

Weightage - 10

Input

```
kufdasciknkascdfd
```

Output

```
kkkuffdddaassccin
```

Weightage - 10

Input

```
psncbjyddsckmpkmakjcac
```

Output

```
ppssnccccbjjyddkkkmmaa
```

Weightage - 15

Input

```
zxascmacnpkjncacjdnqdaphgmakmxasscbas
```

Output

```
zxxaaaaaaassssccccmmmnnnppkkjjddqhgb
```

Weightage - 15

Input

```
mnsakdocnmdnwyevcnbsjdwjhdvcgwecbhjehryvfevhubjgbrygvb
```

Output

```
mmnnnnsssakddddoccccwwwyyyeeeevvvvvbbbbbjjjjhhhhgggrrf
```

Weightage - 15

Input

```
abdscsdghlcndjvcahljxkmnjbsdhgvchlskchdshuclsdjbchdslj
```

Output

```
aabbbbbbddddddddddddsssssssscccccccccgghhhhhhhhhhhlll
```

Weightage - 15

Sample Input

```
abcaaubcc
```

Sample Output

```
aaabbcccu
```

Solution

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

class TestClass {
    public static void printGrouped(char[] str)
    {
        int MAX_CHAR = 26;
        int n = str.length;

        int count[] = new int[MAX_CHAR];

        for (int i = 0 ; i < n ; i++) {
            count[str[i]-'a']++;
        }

        for (int i = 0; i < n ; i++)
        {
            while (count[str[i]-'a'] > 0) {
                System.out.print(str[i]);
                count[str[i]-'a'] = count[str[i]-'a']- 1;
            }
            count[str[i]-'a'] = 0;
        }
    }

    public static void main(String args[] ) throws Exception {
        Scanner in  = new Scanner(System.in);
        String input;
        input = in.nextLine();
        char[] str = input.toCharArray();

        printGrouped(str);
    }
}
```

Q2

Test Case

Input

```
4 2 2
1 4
3 4
```

Output

```
1 1
```

Weightage - 10

Input

```
7 4 6
1 4 6 7
1 2 3 4 6 7
```

Output

```
4 1
```

Weightage - 10

Input

```
5 3 4
1 4 5
1 2 4 5
```

Output

```
3 1
```

Weightage - 10

Input

```
15 10 5
1 2 5 7 8 9 10 11 12 13
1 2 5 12 13
```

Output

```
5 5
```

Weightage - 15

Input

```
12 6 4
1 2 5 6 8 9
1 2 5 12
```

Output

```
3 5
```

Weightage - 15

Input

```
20 15 12
1 2 3 5 6 8 12 13 15 18 19 20
```

Output

```
11 4
```

Weightage - 20

Input

```
26 13 16
1 2 3 4 5 8 9 10 11 20 21 23 26
1 2 3 4 5 8 9 10 11 15 16 18 20 24 25 26
```

Output

```
11 8
```

Weightage - 20

Sample Input

```
7 4 6
1 4 6 7
1 2 3 4 6 7
```

Sample Output

```
4 1
```

Sample Input

```
4 2 2
1 4
3 4
```

Sample Output

```
1 1
```

Solution

```
import java.io.*;
import java.util.*;
class Versionmanagementsystem {
    public static void main(String [] args) {
        int i,j,n,m,k,l,c=0,cc=0;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        m = sc.nextInt();
        k = sc.nextInt();
        int a[] = new int[m];
        int b[] = new int[k];
        for(i=0;i<m;i++) {
            a[i] = sc.nextInt();
        }
        for(i=0;i<k;i++) {
            b[i] = sc.nextInt();
```

```
}
for(i=0;i<m;i++)
{
    for(j=0;j<k;j++)
    {
        if(a[i]==b[j]){c++;}
    }
}
for(i=1;i<=n;i++)
{
    for(j=0;j<m;j++)
    {
        if(a[j]==i)
        {
            break;
        }
        if(j==m-1)
        {
            for(l=0;l<k;l++)
            {
                if(b[l]==i)
                {
                    break;
                }
                if(l==k-1){cc++;}
            }
        }
    }
    System.out.println(c+" "+cc);

 }
}
```

Q3

Test Case

Input	Output
6	********** b*iiiiiii*b bb*iiiii*bb bbb*iii*bbb

Weightage - 10

Input	Output
8	********** b*iiiiiiiii*b bb*iiiiiii*bb bbb*iiiii*bbb

Weightage - 10

Input	Output
10	********** b*iiiiiiiiiii*b bb*iiiiiiiii*bb bbb*iiiiiii*bbb

Weightage - 10

Input	Output
12	

```
**********************
b*iiiiiiiiiiiiiiiiii*b
bb*iiiiiiiiiiiiiiii*bb
bbb*iiiiiiiiiiiiiii*bbb
```

Weightage - 15

Input

```
15
```

Output

```
****************************
b*iiiiiiiiiiiiiiiiiiiiiiii*b
bb*iiiiiiiiiiiiiiiiiiiiii*bb
bbb*iiiiiiiiiiiiiiiiiiii*bbb
```

Weightage - 15

Input

```
18
```

Output

```
**********************************
b*iiiiiiiiiiiiiiiiiiiiiiiiiiii*b
bb*iiiiiiiiiiiiiiiiiiiiiiiii*bb
bbb*iiiiiiiiiiiiiiiiiiiiiiii*bbb
```

Weightage - 20

Input

```
20
```

Output

```
************************************
b*iiiiiiiiiiiiiiiiiiiiiiiiiiiiii*b
bb*iiiiiiiiiiiiiiiiiiiiiiiiiiii*bb
bbb*iiiiiiiiiiiiiiiiiiiiiiiiii*bbb
```

Weightage - 20

Sample Input

```
4
```

Sample Output

```
*******
b*iii*b
bb*i*bb
bbb*bbb
```

Sample Input

```
5
```

Sample Output

```
*********
b*iiiii*b
bb*iii*bb
bbb*i*bbb
```

Solution

```
import java.io.*;
import java.util.*;
class Invertedhollowpyramid {
    public static void main(String [] args) {
        int n,i,j,k;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        k=(n*2)-1;
        for(i=0;i<n;i++)
        {
            for(j=0;j<k;j++)
            {
                if((i==0)||(i==j)||(j==k-i-1)) {
                    System.out.print("*");
                    }
                else if((j<i)||(j>=k-i))
                {
                    System.out.print("b");
```

```
                }
                else
                {
                    System.out.print("i");
                }
            }
            System.out.println();
        }

    }
}
```

Q4

Test Case

Input

```
5
20 0 1 0 2
20 20 0 20 -1
20 20 20 20 20
```

Output

```
Yes
```

Weightage - 10

Input

```
5
20 0 1 0 2
20 20 0 20 -1
20 20 20 20 20
```

Output

```
No
```

Weightage - 10

Input

```
6
20 20 0 20 20 20
3 20 20 6 20 1
20 0 20 20 20 20
```

Output

```
No
```

Weightage - 10

Input

```
9
20 5 20 20 3 20 20 20 20
20 20 8 20 20 20 20 5 20
20 20 20 20 20 20 2 20 20
```

Output

```
No
```

Weightage - 15

Input

```
7
20 0 1 0 2 20 20
20 20 0 20 -1 20 20
20 20 20 20 20 20 20
```

Output

```
Yes
```

Weightage - 15

Input

```
9
20 0 1 0 2 20 20 20 20
20 20 0 20 -1 20 20 20 20
20 20 20 20 20 20 20 20 20
```

Output

```
Yes
```

Weightage - 20

Input

```
12
-1 5 20 20 3 20 20 20 6 20 1 20
20 20 8 20 20 8 20 5 20 1 20 1
20 20 20 20 20 20 2 20 20 20 20 1
```

Output

```
No
```

Weightage - 20

Sample Input

```
5
20 0 1 0 2
20 20 0 20 -1
20 20 20 20 20
```

Sample Output

```
Yes
```

Sample Input

```
5
20 0 1 0 2
20 20 0 20 -1
20 20 20 20 20
```

Sample Output

```
No
```

Solution

```
import java.io.*;
import java.util.*;
class validConfiguration {
    public static void main(String [] args) {
        int i,j,n,s=1;
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        int a[][] = new int[n][n];
        for(i=0;i<n;i++) {
            for(j=0;j<n;j++) {
                a[i][j] = sc.nextInt();
            }
        }
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                if(a[i][j]==1)
                {
                    if((i>0)&&(a[i-1][j]!=0)){s=0;break;}
                    if((i<n-1)&&(a[i+1][j]!=0)){s=0;break;}
                    if((j<n-1)&&(a[i][j+1]!=0)){s=0;break;}
                    if((j>0)&&(a[i][j-1]!=0)){s=0;break;}
            }
            if((a[i][j]>0)&&(a[i][j]<11))
            {
                if((i<n-2)&&(a[i+2][j]<11)&&(a[i+2][j]>0)){if(a[i+1][j]!=0){s=0;break;}}
                if((j<n-2)&&(a[i][j+2]<11)&&(a[i][j+2]>0)){if(a[i][j+1]!=0){s=0;break;}}
                if((i>0)&&(j>0)&&(a[i-1][j-1]<11)&&(a[i-1][j-1]>0)){if((a[i-1][j]!=0)||(a[i][j-1]!=0)){s=0;break;}}
                if((i<n-1)&&(j>0)&&(a[i+1][j-1]<11)&&(a[i+1][j-1]>0)){if((a[i+1][j]!=0)||(a[i][j-1]!=0)){s=0;break;}}
            }
        }
        if(s==0){break;}
    }
    if(s ==0) {
        System.out.println("No");
    }
    else {
        System.out.println("Yes");
    }
```

```
        }
    }
```