

# React Code Test

## Overview

You are tasked with building a simple web application that displays a list of users and their details. You will be provided with a JSON data file that contains a list of users. Your application should be built using React and should be responsive.

## Requirements

1. The web application should display a list of users, with their names, email addresses, and phone numbers.
2. The user should be able to sort the list of users by name, email, or phone number.
3. user should be able to filter the list of users by name, email, or phone number.
4. The web application should be responsive and display properly on desktop and mobile devices.
5. The code should be well-organized and easy to read.

## Data

The JSON data file containing the list of users is available at:

<https://jsonplaceholder.typicode.com/users>

## Guidelines

1. You can use any third-party libraries or tools that you feel are necessary.
2. You should use Git for version control and provide a link to a public repository containing your code.
3. You should provide a README file with instructions on how to run your application.

## Bonus Points

1. Use functional components and hooks
2. Add pagination to the list of users.
3. Use CSS modules or another CSS-in-JS library to style the app
4. Add unit tests to your code.

## Submission

Please provide a link to your code repository and any instructions necessary to run the application.

<https://github.com/bbstc/prizm>

## Java Code Test

1. Write a program to generate the following result.

```
1
12
124
1247
124713
12471324
1247132444
124713244481
124713244481149
.....(and so on)
```

\* Directly print out the result is not acceptable  
(Bonus: If you do not need to use If-Case to handle first few lines)

2. What will be the output of the following code?

```
import java.util.List;
import java.util.ListIterator;
import java.util.ArrayList;

public class ArrayListTest {
    public static void main(String args[]) {
        List<String> arrayList = new ArrayList<String>();
        arrayList.add("a");
        arrayList.add("b");

        ListIterator<String> listIterator = arrayList.listIterator();
        while (listIterator.hasNext()) {
            System.out.println(listIterator.next());
        }
    }
}
```

```

        listIterator.previous();
    }
}

```

### 3. 2D-Grid Maze Game

Assume there have a class Maze, which handle the generation of maze, and provide information for player to determine which direction to move.

In code is something like this:

```

//Initiate a 20x20 maze, with player will start at x:5,y:7
Maze maze = new Maze(20, 5, 7);

```

It provides a function to query what player can do if s/he is at certain position

```

//If player in x:6,y:7, what can s/he do?
List<Maze.Action> actions = maze.PossibleActions(6,7);

```

Maze.Action is an enum, contains following choices:  
Exit, MoveNorth, MoveEast, MoveSouth, MoveWest

Please design a program to determine the exit path from the start position.

(You can answer this question with pseudo code/segments of code to assist the illustration of logic)

### 4. What is the output in Console of the following code?

```

public class WrittenTest {
    public static void main(String[] args) {
        ClassA item1 = new ClassA();
        ClassA item2 = new ClassC();

        assignA(item1, item2);
        assignB(item2, item1);
    }
}

```

```
ClassA item3 = item1;  
item3.setProperty1(item2);
```

```
    printResult(item1);  
    printResult(item2);  
    printResult(item3);  
}
```

```
private static void assignA(ClassA para1, ClassA para2) {  
    para1.setProperty1(para2);  
}
```

```
private static void assignB(ClassA para1, ClassA para2) {  
    para1 = new ClassB();  
    para1.setProperty1(para2);  
}
```

```
private static void printResult(ClassA para1) {  
    if (para1 != null) {  
        if (para1.getProperty1() != null)  
            para1.getProperty1().printResult();  
        else  
            para1.printResult();  
    }  
}  
}
```

```
class ClassA {  
    private ClassA property1;  
  
    public ClassA getProperty1() {  
        return property1;  
    }  
}
```

```

    public void setProperty1(ClassA property1) {
        this.property1 = property1;
    }

    public void printResult() {
        System.out.println("ClassA");
    }
}

class ClassB extends ClassA {
    @Override
    public void printResult() {
        System.out.println("ClassB");
    }
}

class ClassC extends ClassA {
    @Override
    public void printResult() {
        System.out.println("ClassC");
    }
}

```

ClassA  
 ClassC  
 ClassA

5. Please provide SQL statements to output following result tables.

Table : Employ

Emplo yID	Nam e	Deptl D
1	Tom	1
2	Mary	1
3	John	2
4	David	3
5	Jason	3

Table : Department

DeptID	Name
1	SALES
2	HR
3	FINANCE
4	DEV
5	QC

6	Ray	6
---	-----	---

### Result 1

```
SELECT Employ.EmployeeID, Employ.Name,
Department.Name AS DepartmentName
FROM Employ
LEFT JOIN Department ON Employ.DeptID =
Department.DeptID;
```

Emplo yID	Nam e	Department Name
1	Tom	SALES
2	Mary	SALES
3	John	HR
4	David	FINANCE
5	Jason	FINANCE
6	Ray	[NULL]

### Result 2

```
SELECT Department.Name AS
DepartmentName,
COUNT(Employ.EmployeeID) AS [No Of
Staffs]
FROM Department
LEFT JOIN Employ ON Department.DeptID =
Employ.DeptID
GROUP BY Department.Name;
```

DepartmentNam e	No Of Staffs
SALES	2
HR	1
FINANCE	2
DEV	0
QC	0

### Result 3

```
SELECT Department.Name AS
DepartmentName,
COUNT(Employ.EmployeeID) AS [No Of Staffs]
FROM Department
LEFT JOIN Employ ON Department.DeptID =
Employ.DeptID
GROUP BY Department.Name
UNION
SELECT 'Unknown' AS DepartmentName,
COUNT(*) AS [No Of Staffs]
FROM Employ
WHERE DeptID IS NULL;
```

DepartmentNam e	No Of Staffs
SALES	2
HR	1
FINANCE	2

DEV	0
QC	0
Unknown	1