**Bài tập Java/TestNG**

Input Data:

* 4 Arrays:

1. String [] Array\_1 = {};
2. String [][] Array\_2 = {};
3. String [] Array\_3 = {"1D\_Value1", "1D\_Value2", "1D\_Value3", "1D\_Value4", "1D\_Value5"};
4. String [][] Array\_4 = {{"2D\_1\_Value1", "2D\_1\_Value2"},

{"2D\_2\_Value1", "2D\_2\_Value2", "2D\_2\_Value3"},

{"2D\_3\_Value1", "2D\_3\_Value2", "2D\_3\_Value3", "2D\_3\_Value4", "2D\_3\_Value5"}};

* 10 Strings:

1. String str\_1 = "Wrong String Format";
2. String str\_2 = "Start getData ${\_getData[1]}...";
3. String str\_3 = "Start getData ${\_getData[6]}...";
4. String str\_4 = "Start getData ${\_getData[a%c]}...";
5. String str\_5 = "Start getData ${\_getData[4][2]}.....";
6. String str\_6 = "Start getData ${\_getData[1][2]}.....";
7. String str\_7 = "Start getData ${\_getData[0][3]}.....";
8. String str\_8 = "Start getData ${\_getData[1%c][2]}.....";
9. String str\_9 = "Start getData ${\_getData[0][wrong index]}.....";
10. String str\_10 = "Start getData ${\_getData[a%3][wrong index]}.....";
11. String str\_10 = "Start getData ${\_getData[0]} and ${\_getData[1]}.....";
12. String str\_10 = "Start getData ${\_getData[0]} and ${\_getData[10]}.....";
13. String str\_10 = "Start getData ${\_getData[0]} and ${\_getData[1\*ac]}.....";
14. String str\_10 = "Start getData ${\_getData[10]} and ${\_getData[1\*ac]}.....";
15. String str\_10 = "Start getData ${\_getData[0][0]} and ${\_getData[1][0]}.....";
16. String str\_10 = "Start getData ${\_getData[0][0]} and ${\_getData[10][0]}.....";
17. String str\_10 = "Start getData ${\_getData[0][0]} and ${\_getData[1][10]}.....";
18. String str\_10 = "Start getData ${\_getData[0][0]} and ${\_getData[a\*c][12a]}.....";
19. String str\_10 = "Start getData ${\_getData[10][10]} and ${\_getData[a\*c][12a]}.....";
20. String str\_10 = "Start getData ${\_getData[0]} and ${\_getData[1][1]}.....";
21. String str\_10 = "Start getData ${\_getData[10]} and ${\_getData[1][1]}.....";
22. String str\_10 = "Start getData ${\_getData[0]} and ${\_getData[10][1]}.....";
23. String str\_10 = "Start getData ${\_getData[0]} and ${\_getData[1][10]}.....";
24. String str\_10 = "Start getData ${\_getData[abc]} and ${\_getData[1][1]}.....";
25. String str\_10 = "Start getData ${\_getData[abc]} and ${\_getData[1%wr][wr]}.....";
26. String str\_10 = "Start getData ${\_getData[1]} and ${\_getData[0][1]}, ${\_getData[2]}, ${\_getData[0]} and ${\_getData[2][1]}.....";

**Total 72 Test Cases, expected result: 23 PASSED and 49 FAILED.**

* Test Array:

1. Is 1-Dimensional Array?
2. Is 2-Dimensional Array?
3. Is Empty 1-Dimensional Array?
4. Is Empty 2-Dimensional Array?

* Test String:

1. Wrong format. (Does not contains string "${\_getData[i]}" or “${\_getData[i][j]})
2. Correct format:
3. Contains string “"${\_getData[i]}”: (Get data of 1-Dimensional Array)
4. i is Integer:
5. i < Array.length
6. i >= Array.length
7. i is not Integer
8. Contains string “${\_getData[i][j]}”: (Get data of 1-Dimensional Array)
9. i & j are Integer:
10. i > Array.length
11. i < Array.length && j < Array[i].length
12. i < Array.length && j >= Array[i].length
13. i is not Integer.
14. j is not Integer.
15. I & j are not Integer.
16. ……………………..

**Scenarios:**

1. 1 – A => Fail
2. 1 – B => Fail
3. 2 – A => Fail
4. 2 – B => Fail
5. 3 – A => Fail
6. 3 – B\_i\_a)\_11 => True => print the value of Array[i].
7. 3 – B\_i\_a)\_12 => Fail
8. 3 – B\_i\_b) => Fail
9. 3 – B\_ii => Fail
10. 4 – A => Fail
11. 4 – B\_i\_a)\_11 => True => print the value of Array[i].
12. 4 – B\_i\_a)\_12 => Fail
13. 4 – B\_i\_b) => Fail
14. 4 – B\_ii\_a)\_11 => Fail
15. 4 – B\_ii\_a)\_12 => True => print the value of Array[i][j].
16. 4 – B\_ii\_a)\_13 => Fail
17. 4 – B\_ii\_b) => Fail
18. 4 – B\_ii\_c) => Fail
19. 4 – B\_ii\_d) => Fail

………………………..

**Test Case:**

1. TC1 -- Test with Empty 1-Dimensional Array and Wrong String.
2. TC2 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is valid & i < length of 1-Dimensional Array)
3. TC3 -- Test with Empty 2-Dimensional Array and Wrong String
4. TC4 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are valid)
5. TC5 -- Test with 1-Dimensional Array and Wrong String
6. TC6 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is valid) => Return 1 value
7. TC7 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is valid && i >= length of 1-Dimensional Array)
8. TC8 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is invalid)
9. TC9 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]})
10. TC10 -- Test with 2-Dimensional Array and Wrong String
11. TC11 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is valid && i < length of 2-Dimensional Array) => Return 1 value
12. TC12 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is valid && i >= length of 2-Dimensional Array)
13. TC13 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is invalid)
14. TC14 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is valid && i >= length of 2-Dimensional Array)
15. TC15 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are valid) => Return 1 value
16. TC16 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are valid and j > length of 2-Dimensional Array[i])
17. TC17 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is invalid)
18. TC18 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and j is invalid)
19. TC19 -- Test with 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are invalid)

Test Case Extend:

1. TC20 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is valid & i >= length of 1-Dimensional Array)
2. TC21 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is invalid)
3. TC22 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are invalid and i >= length of 1-Dimensional Array)
4. TC23 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are valid)
5. TC24 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are valid and j >= length of 1- Dimensional Array[i])
6. TC25 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is invalid and j is valid)
7. TC26 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is valid and j is invalid)
8. TC27 -- Test with Empty 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are invalid)
9. TC28 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is valid)
10. TC29 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is valid & i >= length of 2-Dimensional Array)
11. TC30 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i]} and i is invalid)
12. TC31 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are invalid and i >= length of 2-Dimensional Array)
13. TC32 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are valid and j >= length of 2- Dimensional Array[i])
14. TC33 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is invalid and j is valid)
15. TC34 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is valid and j is invalid)
16. TC35 -- Test with Empty 2-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are invalid)
17. TC36 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i >= length of 1-Dimensional Array)
18. TC37 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and j >= length of 1-Dimensional Array[i])
19. TC38 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is invalid & j is valid)
20. TC39 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i is valid & j is invalid)
21. TC40 -- Test with 1-Dimensional Array and Correct String (String contains ${\_getData[i][j]} and i & j are invalid)

Test Cases for Multiple Correct Format in String:

1. TC41 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 1D Array, both are valid) => Return 2 values
2. TC42 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 1D Array, 1 is valid and 1 have i > len(1D Array)) => Return 1 value
3. TC43 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 1D Array, 1 is valid and 1 is invalid) => Return 1 value
4. TC44 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 1D Array, 1 > len(1D Array) and 1 is invalid)
5. TC45 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, both are valid)
6. TC46 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, 1 is valid and 1 have i > len(Array))
7. TC47 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, 1 is valid and 1 have j > len(Array[i]))
8. TC48 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, 1 is valid and is invalid)
9. TC49 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, 1 have i and j > len(Array) and is invalid)
10. TC50 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, both are valid) => Return 1 value
11. TC51 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, 1 have i > len(1D Array) and 1 is valid)
12. TC52 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, 1 is valid and 1 have i > len(2D Array) => Return 1 value
13. TC53 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, 1 is valid and 1 have j > len(2D Array[i]) => Return 1 value
14. TC54 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, 1 is invalid and 1 is valid)
15. TC55 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, both are invalid)
16. TC56 -- Test with 1-Dimensional Array and 2 Correct Format in String (Get 3 data of 1D Array and get 2 data of 2D Array, all are valid) => Return 3 values
17. TC57 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 1D Array, both are valid) => Return 2 values
18. TC58 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 1D Array, 1 is valid and 1 > len(1D Array)) => Return 1 value
19. TC59 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 1D Array, 1 is valid and 1 is invalid) => Return 1 value
20. TC60 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 1D Array, 1 > len(1D Array) and 1 is invalid)
21. TC61 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, both are valid) => Return 2 values
22. TC62 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, 1 is valid and 1 have i > len(Array)) => Return 1 value
23. TC63 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, 1 is valid and 1 have j > len(Array[i])) => Return 1 value
24. TC64 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, 1 is valid and is invalid) => Return 1 value
25. TC65 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 2 data of 2D Array, 1 have i and j > len(Array) and is invalid)
26. TC66 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, both are valid) => Return 2 values
27. TC67 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, 1 have i > len(2D Array) and 1 is valid) => Return 1 value
28. TC68 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, 1 is valid and 1 have i > len(2D Array) => Return 1 value
29. TC69 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, 1 is valid and 1 have j > len(2D Array[i]) => Return 1 value
30. TC70 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, 1 is invalid and 1 is valid) => Return 1 value
31. TC71 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 1 data of 1D Array and get 1 data of 2D Array, both are invalid)
32. TC72 -- Test with 2-Dimensional Array and 2 Correct Format in String (Get 3 data of 1D Array and get 2 data of 2D Array, all are valid) => Return 5 values