

Data Structures and Algorithms

Training Exam

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	20/11/2012

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	18/Apr/2019	Add new	Add	DieuNT1	VinhNV

Contents

Problem 01: Rotate Array	5
Specifications:	5
Technical Requirements for Java:	6
Technical Requirements for .NET:.....	6
Technical Requirements for JavaScript:	6
Problem 02: Search.....	7
Specifications:	7
Technical Requirements for Java:	8
Technical Requirements for .NET:.....	8
Technical Requirements for JavaScript:	9
Problem 03: Array Multiplication	9
Specifications:	9
Technical Requirements for Java:	10
Technical Requirements for .NET:.....	10
Technical Requirements for JavaScript:	11

CODE:	DSA_Practice_T102
TYPE:	Long
LOC:	N/A
DURATION:	180 MINUTES

Require 001: Working tools and Delivery requirements

- **Working tools:** Eclipse IDE for Java **OR** Visual Studio for .net
- **Delivery:** Source code and test results in a compressed archive.

Require 002: Technologies

The product illustrates:

- Data structures: Array, Stack, Queue, Singly Linked List
- Sorting algorithms
- Searching algorithms

Require 002: Technical Requirements

- Follow the standard naming and coding convention.
- Add appropriate comments for each class, method, attribute, algorithms, ...
- Use console application template
- Create Project/Solution and package/namespace

Student is requested to create project solution to resolve problems bellow.

For Java: create project named: DSA.Practice.T102. Each problem, create 1 package for it.

For C#: create solution named: DSA.Practice.T102. Each problem, create 1 project for it.

For JavaScript: create project named: dsa.practice.t102. Each problem, create 1 package for it.

Problem 01: Rotate Array

Specifications:

Give an array of positive numbers. Find the maximum of distance between 2 consecutive numbers.

Distance between 2 numbers is the absolute different of those.

If the array is null or empty then return an empty array.

Example 1:

Input:

```
arrays: { 1, 2, 3, 4, 5 }  
n: 1
```

Output:

```
{ 5, 1, 2, 3, 4 }
```

Explanation: Every number is shift to the right by 1 so 1 take place and 2, 2 takes place of 3 and so on. 5 is the last element so I will take the place of 1

Example 2:

Input:

```
arrays: { 1, 2, 3, 4, 5 }  
n: 3
```

Output:

```
{ 3, 4, 5, 1, 2 }
```

Example 3:

Input:

```
arrays: { 1, 2, 3, 4, 5 }  
n: -3
```

Output:

```
{ 1, 2, 3, 4, 5 }
```

Example 4:

Input:

```
arrays: null  
n: 1
```

Output:

```
{ }
```

Example 5:**Input:**

```
arrays: { }  
n: 1
```

Output:

```
{ }
```

Technical Requirements for Java:

- Create a class called **Problem01** in package **fa.training.dsa.test02.option01**
- In **Problem01** class, create the method with signature below:

Input:

- ✓ An array of positive numbers.
- ✓ An integer

Output:

- ✓ An array of integer

```
public static int[] rotateArray(int[] arrays, int n) {}
```

- Must test your code before submit

Technical Requirements for .NET:

- Create project named **DSA.Practice.T102.Problem1** inside the solution
- Create a class called **Problem01** in the project
- In **Problem01** class, create the method with signature below:

Input:

- ✓ An array of positive numbers.
- ✓ An integer

Output:

- ✓ An array of integer

```
public static int[] RotateArray(int[] arrays, int n) {}
```

- Must test your code before submit

Technical Requirements for JavaScript:

- Create a file called **problem01.js** in folder **fa/training/dsa/test02/option02**
- In **problem01.js** file, create the function with signature below:

Input:

- ✓ An array of positive numbers.
- ✓ An integer

Output:

- ✓ An array of integer

```
Function rotateArray(arrays, n) {}
```

- Must test your code before submit

Problem 02: Search**Specifications:**

Given an integer array, if an integer p exists in the array such that the count of elements greater than p in the array, that number equals to p .

If such an integer is found return true else return false.

If the array is null or empty return false.

Example 1:**Input:**

```
arrays: { 1, 6, 7, 2 }
```

Output:

```
true
```

Explanation:

- There are 3 numbers (6, 7, 2) greater than 1
- There are 1 number (7) greater than 6
- There is no number greater than 7
- There are 2 numbers (6, 7) greater than 2

Example 2:**Input:**

```
arrays: { 1, 2, 7, 0, 9, 3, 6, 0, 6 }
```

Output:

```
false
```

Explanation:

- There are 7 numbers (1, 2, 7, 9, 3, 6, 6) greater than 0
- There are 6 numbers (2, 7, 9, 3, 6, 6) greater than 1
- There are 5 numbers (7, 9, 3, 6, 6) greater than 2
- There are 4 numbers (7, 9, 6, 6) greater than 3
- There are 2 numbers (7, 9) greater than 6
- There is 1 number (9) greater than 7
- There is no number greater than 9

Example 3:**Input:**`arrays: { }`**Output:**`false`**Example 4:****Input:**`arrays: null`**Output:**`false`**Technical Requirements for Java:**

- Must create a class called **Problem02** in package **fa.training.dsa.test02.option02**
- Must create a method name **checkArray** with below signature

Input:

- An array of integer

Output:

- A boolean true or false
- The method must have below signature:

```
public static boolean checkArray(int[] arrays) {}
```

- Must test your code before submit

Technical Requirements for .NET:

- Create project named **DSA.Practice.T102.Problem2** inside the solution
- Must create a class called **Problem02** in the project
- Must create a method name **CheckArray** with below signature.

Input:

- An array of integer

Output:

- A Boolean true or false
- The method must have below signature:

```
public static bool CheckArray(int[] arrays) {}
```

- Must test your code before submit

Technical Requirements for JavaScript:

- Must create a file called **problem02.js** in folder **fa/training/dsa/test02/option02**
- Must create a function called **checkArray** with below signature

Input:

- An array of integer

Output:

- A boolean: true or false
- The function must have below signature:

```
export function checkArray(int[] arrays) {}
```

- Must test your code before submit

Problem 03: Array Multiplication**Specifications:**

Given 2 arrays of number A and B (both represents an integer number).

For example: A = {1, 2, 3} represent the number 123

Calculate the multiplication of numbers that those 2 arrays represented.

The return valued must be an array of number.

If either array A or B is empty or null, then return an empty array

Example 1:

Input:

A: { 1, 2, 3 }

B: { 1, 0 }

Output:

{ 1, 2, 3, 0 }

Explain:

A represents the number 123

B represents the number 10

$A * B = 123 * 10 = 1230$

Hence, we return { 1, 2, 3, 0 }

Example 2:

Input:

A: { 1, 2, 3, 4, 5, 6, 7, 8, 9 }

B: { 1, 2, 3, 4, 5, 6, 7, 8, 9, 1 }

Output:

```
{ 1, 5, 2, 4, 1, 5, 7, 8, 7, 6, 2, 5, 3, 6, 1, 9, 9, 9 }
```

Explain: Since $123456789 * 1234567891 = 152415787625361999$

So we return the above array

Example 3:

Input:

```
A: { }
```

```
B: { 1, 2, 3 }
```

Output:

```
{ }
```

Example 4:

Input:

```
A: null
```

```
B: { 1, 2, 3 }
```

Output:

```
{ }
```

Note 1: The numbers can be arbitrarily large and are non-negative.

Note 2: Your answer should not have leading zeroes. For example, { 0, 1, 0, 0 } is not a valid answer.

Technical Requirements for Java:

- Must create a class called **Problem03** in package **fa.training.dsa.test02.option03**
- Must create the following method in class **Problem03**:

```
public static int[] multiplyArrays(int[] a, int[] b) {}
```

- Must complete the method above to fulfil the requirement
- Must NOT use 3rd library nor standard library
- Must NOT join the array and convert to number then do the multiplication, it will give wrong answer on very large number.

Technical Requirements for .NET:

- Create project named **DSA.Practice.T102.Problem3** inside the solution
- Must create a class called **Problem03** in the project
- Must create the following method in class **Problem03**:

```
public static int[] MultiplyArrays(int[] a, int[] b) {}
```

- Must complete the method above to fulfil the requirement

- Must NOT use 3rd library nor standard library
- Must NOT join the array and convert to number then do the multiplication, it will give wrong answer on very large number.

Technical Requirements for JavaScript:

- Must create a file called **problem03.js** in folder **fa/training/dsa/test02/option03**
- Must create the following function in file **problem03.js**:

```
export function multiplyArrays(a, b) {}
```

- Must complete the method above to fulfil the requirement
- Must NOT use 3rd library nor standard library
- Must NOT join the array and convert to number then do the multiplication, it will give wrong answer on very large number.

--THE END--