

## DAILY ONLINE ACTIVITIES SUMMARY

<b>Date:</b>	02/07/2020	<b>Name:</b>	Prajwal
<b>Sem &amp; Sec</b>	IV sem & B sec	<b>USN:</b>	4AL18CS057
<b>Online Test Summary</b>			
<b>Subject</b>	-----		
<b>Max. Marks</b>	-----	<b>Score</b>	-----
<b>Certification Course Summary</b>			
<b>Course</b>	Python For Data Science		
<b>Certificate Provider</b>	COGNITIVE CLASS	<b>Duration</b>	12 hours
<b>Coding Challenges</b>			
<b>Problem Statement:</b> 1. Write a Java Program minimize the maximum difference between adjacent elements in an array.			
<b>Status:</b> Done			
<b>Uploaded the report in Github</b>		YES	
<b>If yes Repository name</b>		<a href="https://github.com/PRAJWALKOTIAN/lockdown-coding">https://github.com/PRAJWALKOTIAN/lockdown-coding</a>	
<b>Uploaded the report in slack</b>		YES	

### **Online test details**

No test was conducted dated on 02 july 2020.

## Certification Course Details

The course I have chosen is python for data science in this I studied the simple APIs as main topic.

4G 10:03 203 KB/s

Voice LTE 45%

[Course](#) > [Modul...](#) > [API's](#) > Simple ...

<

>

### Simple APIs Part 2

[Bookmark this page](#)

### Video

```
graph LR; C1((Client 1)) -- "API inputs Data" --> WS((Web Service)); WS -- "outputs" --> C1;
```

1:16 / 5:01

HD

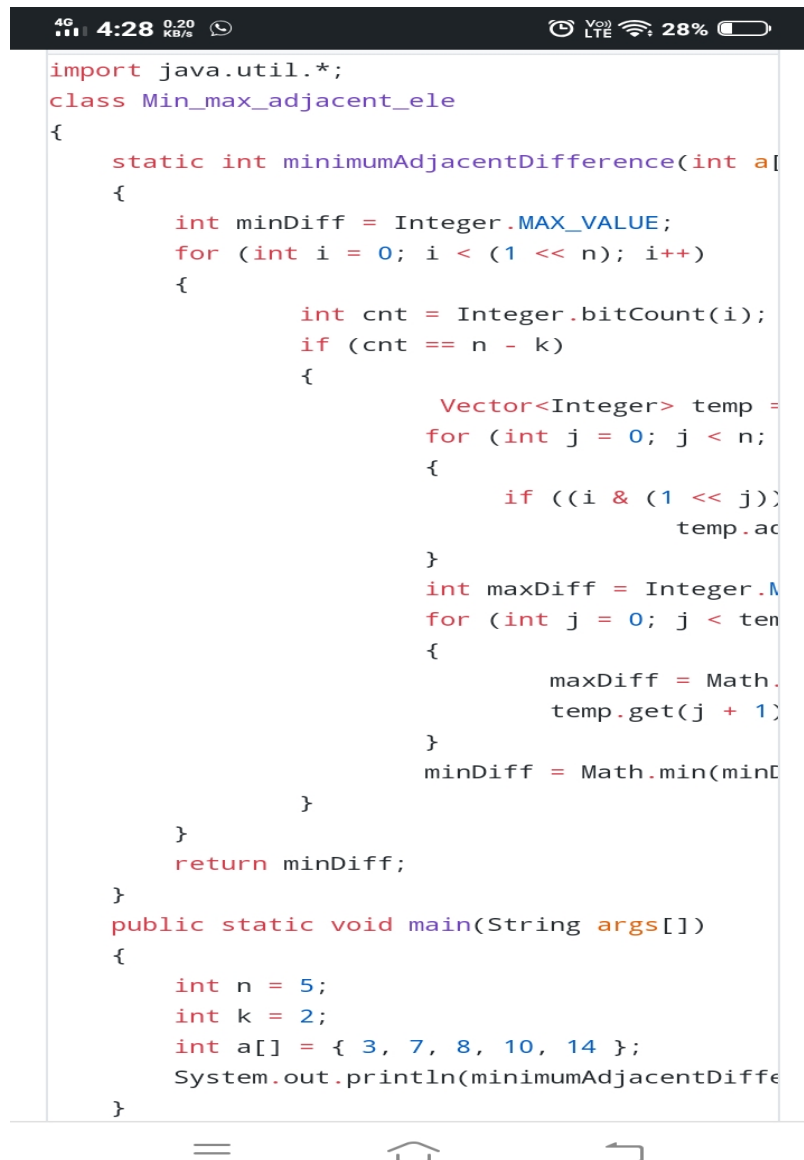
< Previous

Next >

## Coding Challenges Details

The bellow given codes are there on my github repository <https://github.com/PRAJWALKOTIAN/lockdown-coding>

1. Write a Java Program minimize the maximum difference between adjacent elements in an array.



```
import java.util.*;
class Min_max_adjacent_ele
{
    static int minimumAdjacentDifference(int a[]
    {
        int minDiff = Integer.MAX_VALUE;
        for (int i = 0; i < (1 << n); i++)
        {
            int cnt = Integer.bitCount(i);
            if (cnt == n - k)
            {
                Vector<Integer> temp =
                for (int j = 0; j < n;
                {
                    if ((i & (1 << j))
                        temp.add(a[j]);
                }
                int maxDiff = Integer.MAX_VALUE;
                for (int j = 0; j < temp.size(); j++)
                {
                    maxDiff = Math.max(maxDiff,
                        temp.get(j) - temp.get(j - 1));
                }
                minDiff = Math.min(minDiff, maxDiff);
            }
        }
        return minDiff;
    }
    public static void main(String args[])
    {
        int n = 5;
        int k = 2;
        int a[] = { 3, 7, 8, 10, 14 };
        System.out.println(minimumAdjacentDifference(a, n, k));
    }
}
```