#TWO SUM

Two Sum II - Input array is sorted

Given a sorted array of integers, return the indices of the two numbers such that they add up to a specific target.

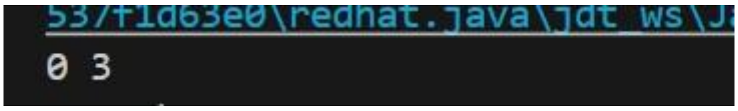
Input:

{-4,0,7,2}

Target:3

Output:

0 3



```
537f1d63e0\redhat.java\jdt_ws\J
0 3
```

#Subarray Sum Equals K

Given an array of integers and a target sum k, return the total number of continuous subarrays whose sum equals to k.

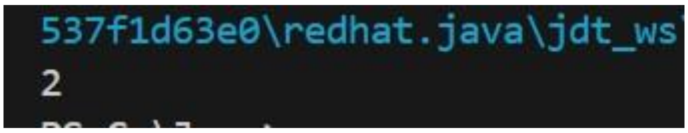
Input:

arr={1,2,8,10}

target=10

Output:

2



```
537f1d63e0\redhat.java\jdt_ws
2
```

Day 1

#Long substring without repeating characters

Input:

Qwertyuuu

Ouput:

7

```
dt_ws\Training
7
```

#RemoveNth

Remove Nth Node From End of List

Given a linked list, remove the nth node from the end and return its head.

Input:

List:

1 9 0 2

n=1

Output:

1 9 0

```
537f1d63e0\redhat.java\jdt_
1 9 0
PS C:\Java>
```

#AddIntegerInLinkedLists

You are given two non-empty linked lists representing two non-negative integers. Add the two numbers and return the sum as a linked list.

Input:

List1:9 9 9

List2:0 0 1 Output:

1 0 0 0

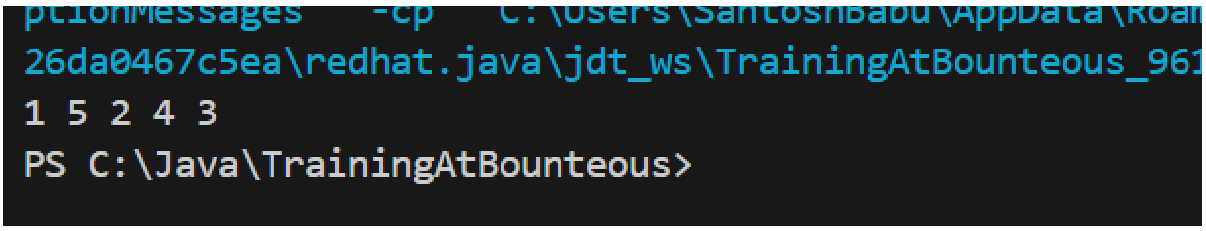
```
537f1d63e0\redhat.java\jdt_ws
1 0 0 0
```

#ReOrder LinkedList
Reorder a linked list from L0 → L1 → … → Ln-1 → Ln to L0 → Ln → L1 → Ln-1 → L2 → Ln-2 → ….

Input:

1 2 3 4 5

Output:
1 5 2 4 3



```
ptionMessages  -cp  C:\Users\SantoshBabu\AppData\Roam
26da0467c5ea\redhat.java\jdt_ws\TrainingAtBounteous_961
1 5 2 4 3
PS C:\Java\TrainingAtBounteous>
```

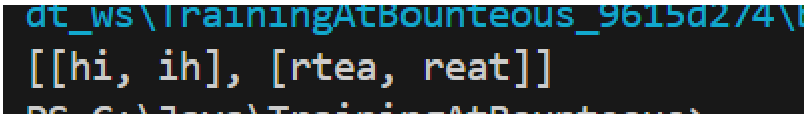
#Group Anagrams

Given an array of strings, group the anagrams together.

Input:

Arr={hi,reat,ih,rtea}

Output:



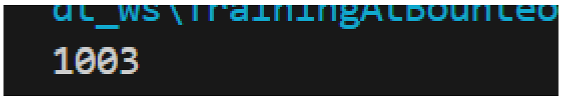
```
dt_ws\TrainingAtBounteous_9615d274\
[[hi, ih], [rtea, reat]]
```

#Rearrange a no to find min possible no in o(n) and constant space.

Input:

3010

Output:



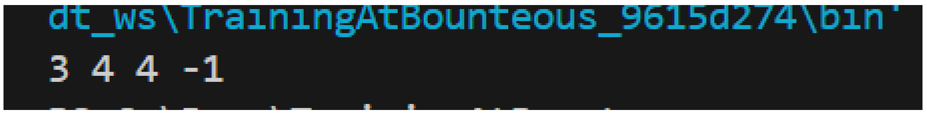
```
dt_ws\TrainingAtBounteo
1003
```

#Next Greater Element
Given a circular array, find the next greater number for every element.

Input:

1 3 2 4

Output:



```
dt_ws\TrainingAtBounteous_9615d274\bin
3 4 4 -1
```

Day2

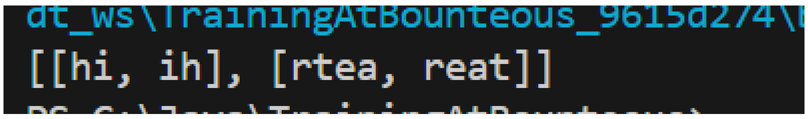
#Group Anagrams

Given an array of strings, group the anagrams together.

Input:

Arr={hi,reat,ih,rtea}

Output:



```
dt_ws\TrainingAtBounteous_9615d274\
[[hi, ih], [rtea, reat]]
PS C:\Java\TrainingAtBounteous>
```

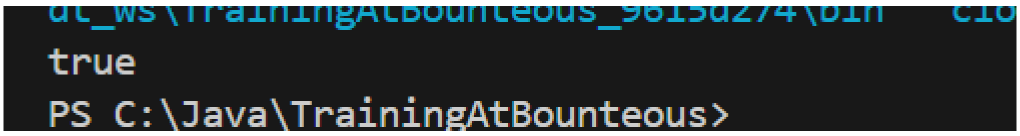
#Close Strings

Two strings are considered close if you can swap letters or change the frequency of any letter to match the other string. Determine if two given strings are close.

Input:

abc, bca

Output:



```
dt_ws\TrainingAtBounteous_9615d274\bin   clo
true
PS C:\Java\TrainingAtBounteous>
```

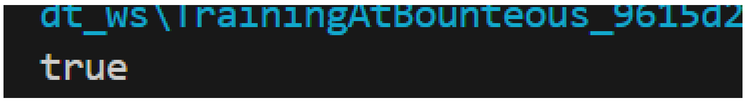
#Valid Parenthesis

Given a string containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

Input:

({[]})

Output:



```
dt_ws\TrainingAtBounteous_9615d2
true
```