
Started on Tuesday, 28 October 2025, 8:04 PM

State Finished

Completed on Tuesday, 28 October 2025, 8:49 PM

Time taken 45 mins 26 secs

Marks 10.00/10.00

Grade **100.00** out of 100.00

Question 1 | Correct Mark 1.00 out of 1.00

Balanced strings are those that have an equal quantity of 'L' and 'R' characters.

Given a balanced string *s*, split it in the maximum amount of balanced strings.

Return the maximum amount of split balanced strings.

Example 1:

Input:

RLRRLRLRL

Output:

4

Explanation: *s* can be split into "RL", "RRL", "RL", "RL", each substring contains same number of 'L' and 'R'.

Example 2:

Input:

RLLLLRRRLR

Output:

3

Explanation: *s* can be split into "RL", "LLLLRRR", "LR", each substring contains same number of 'L' and 'R'.

Example 3:

Input:

LLLLRRRR

Output:

1

Explanation: *s* can be split into "LLLLRRRR".

Constraints:

$1 \leq s.length \leq 1000$

s[*i*] is either 'L' or 'R'.

s is a balanced string.

For example:

Test	Result
<code>print(BalancedStrings('RLRRLRLRL'))</code>	4
<code>print(BalancedStrings('RLLLLRRRLR'))</code>	3

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 def BalancedStrings(s):
2     c=0
3     b=0
4     for char in s:
5         if char=='L':
6             b+=1
7         else:
8             b-=1
```

```
9 |         if b==0:
10 |             c+=1
11 |     return c
```

	Test	Expected	Got	
✓	print(BalancedStrings('RLRLLRLRL'))	4	4	✓
✓	print(BalancedStrings('RLLLLRRRLR'))	3	3	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 2 | Correct Mark 1.00 out of 1.00

You are given an $m \times n$ integer matrix `matrix` with the following two properties:

- Each row is sorted in non-decreasing order.
- The first integer of each row is greater than the last integer of the previous row.

Given an integer `target`, return `True` if `target` is in `matrix` or `False` otherwise.

You must write a solution in $O(\log(m * n))$ time complexity.

Example 1:

1	3	5	7
10	11	16	20
23	30	34	60

Input: `matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]]`, `target = 3`

Output: `True`

Example 2:

1	3	5	7
10	11	16	20
23	30	34	60

Input: `matrix = [[1,3,5,7],[10,11,16,20],[23,30,34,60]]`, `target = 13`

Output: `False`

For example:

Test	Result
<code>print(searchMatrix([[1,3,5,7],[10,11,16,20],[23,30,34,60]], 13))</code>	False
<code>print(searchMatrix([[1,3,5,7],[10,11,16,20],[23,30,34,60]], 3))</code>	True

Answer: (penalty regime: 0 %)

Reset answer

```

1 def searchMatrix(matrix: list[list[int]], target: int) -> bool:
2     if not matrix or not matrix[0]:
3         return False
4     rows=len(matrix)
5     cols=len(matrix[0])
6     row = 0

```

```

6     row=0
7     col=cols-1
8     while row<rows and col>=0:
9         if matrix[row][col]==target:
10             return True
11         elif matrix[row][col]>target:
12             col-=1
13         else:
14             row+=1
15     return False
16

```

	Test	Expected	Got	
✓	print(searchMatrix([[1,3,5,7],[10,11,16,20],[23,30,34,60]], 13))	False	False	✓
✓	print(searchMatrix([[1,3,5,7],[10,11,16,20],[23,30,34,60]], 3))	True	True	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 3 | Correct Mark 1.00 out of 1.00

An list contains N numbers and you want to determine whether two of the numbers sum to a given number K. For example, if the input is 8, 4, 1, 6 and K is 10, the answer is yes (4 and 6). A number may be used twice.

Input Format

The first line contains a single integer n , the length of list

The second line contains n space-separated integers, list[i].

The third line contains integer k.

Output Format

Print Yes or No.

Sample Input

```
7
0 1 2 4 6 5 3
1
```

Sample Output

```
Yes
```

For example:

Input	Result
5 8 9 12 15 3 11	Yes
6 2 9 21 32 43 43 1 4	No

Answer: (penalty regime: 0 %)

```
1 def has_pair_with_sum(lst,k):
2     seen=set()
3     for num in lst:
4         if(k-num) in seen:
5             return "Yes"
6         seen.add(num)
7     return "No"
8 n=int(input())
9 lst=list(map(int,input().split()))
10 k=int(input())
11 print(has_pair_with_sum(lst,k))
```

	Input	Expected	Got	
✓	5 8 9 12 15 3 11	Yes	Yes	✓
✓	6 2 9 21 32 43 43 1 4	No	No	✓
✓	6 13 42 31 4 8 9 17	Yes	Yes	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 4 | Correct Mark 1.00 out of 1.00

Given two Strings s1 and s2, remove all the characters from s1 which is present in s2.

Constraints

1<= string length <= 200

Sample Input 1

experience
enc

Sample Output 1

xpri

Answer: (penalty regime: 0 %)

```
1 s1=input().strip()
2 s2=input().strip()
3 result=''.join([ch for ch in s1 if ch not in s2])
4 print(result)
```

	Input	Expected	Got	
✓	experience enc	xpri	xpri	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 5 | Correct Mark 1.00 out of 1.00

Given an list, find peak element in it. A peak element is an element that is greater than its neighbors.

An element $a[i]$ is a peak element if

$A[i-1] \leq A[i] \geq a[i+1]$ for middle elements. $[0 < i < n-1]$

$A[i-1] \leq A[i]$ for last element $[i=n-1]$

$A[i] \geq A[i+1]$ for first element $[i=0]$

Input Format

The first line contains a single integer n , the length of A .

The second line contains n space-separated integers, $A[i]$.

Output Format

Print peak numbers separated by space.

Sample Input

5

8 9 10 2 6

Sample Output

10 6

For example:

Input	Result
4 12 3 6 8	12 8

Answer: (penalty regime: 0 %)

```
1 n=int(input())
2 a=list(map(int,input().split()))
3 print(*[a[i] for i in range(n) if (i==0 and a[i]>a[i+1]) or (i==n-1 and a[i]>a[i-1]) or (0<i<n-1 and a[i]>a[i-1])])
```

	Input	Expected	Got	
✓	7 15 7 10 8 9 4 6	15 10 9 6	15 10 9 6	✓
✓	4 12 3 6 8	12 8	12 8	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 6 | Correct Mark 1.00 out of 1.00

Given an array `nums` containing `n` distinct numbers in the range `[0, n]`, return *the only number in the range that is missing from the array*.

Example 1:

Input: `nums = [3,0,1]`

Output: 2

Explanation: `n = 3` since there are 3 numbers, so all numbers are in the range `[0,3]`. 2 is the missing number in the range since it does not appear in `nums`.

Example 2:

Input: `nums = [0,1]`

Output: 2

Explanation: `n = 2` since there are 2 numbers, so all numbers are in the range `[0,2]`. 2 is the missing number in the range since it does not appear in `nums`.

Example 3:

Input: `nums = [9,6,4,2,3,5,7,0,1]`

Output: 8

Explanation: `n = 9` since there are 9 numbers, so all numbers are in the range `[0,9]`. 8 is the missing number in the range since it does not appear in `nums`.

For example:

Test	Result
<code>print(missingNumber([3,0,1]))</code>	2
<code>print(missingNumber([0,1]))</code>	2

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 def missingNumber(numbers):
2     n=len(numbers)
3     exp=n*(n+1)//2
4     act=sum(numbers)
5     return exp-act
```

	Test	Expected	Got	
✓	print(missingNumber([3,0,1]))	2	2	✓
✓	print(missingNumber([0,1]))	2	2	✓
✓	print(missingNumber([9,6,4,2,3,5,7,0,1]))	8	8	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 7 | Correct Mark 1.00 out of 1.00

Write a Python program for binary search.

For example:

Input	Result
1,2,3,5,8 6	False
3,5,9,45,42 42	True

Answer: (penalty regime: 0 %)

```

1 def binary_search(arr,tar):
2     arr.sort()
3     low=0
4     high=len(arr)-1
5     while low<=high:
6         mid=(low+high)//2
7         if arr[mid]==tar:
8             print("True")
9             return
10        elif arr[mid]<tar:
11            low=mid+1
12        else:
13            high=mid-1
14        print("False")
15 arr=list(map(int,input().split(",")))
16 tar=int(input())
17 binary_search(arr,tar)

```

	Input	Expected	Got	
✓	1,2,3,5,8 6	False	False	✓
✓	3,5,9,45,42 42	True	True	✓
✓	52,45,89,43,11 11	True	True	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 8 | Correct Mark 1.00 out of 1.00

String should contain only the words are not palindrome.

Sample Input 1

Malayalam is my mother tongue

Sample Output 1

is my mother tongue

Answer: (penalty regime: 0 %)

```
1 | print(' '.join(w for w in input().split() if w.lower() != w[::-1].lower()))
```

	Input	Expected	Got	
✓	Malayalam is my mother tongue	is my mother tongue	is my mother tongue	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 9 | Correct Mark 1.00 out of 1.00

Given an array of integers `nums` which is sorted in ascending order, and an integer `target`, write a function to search `target` in `nums`.

If `target` exists, then return its index. Otherwise, return `-1`.

You must write an algorithm with $O(\log n)$ runtime complexity.

Example 1:

Input: `nums = [-1,0,3,5,9,12]`, `target = 9`

Output: `4`

Explanation: 9 exists in `nums` and its index is 4

Example 2:

Input: `nums = [-1,0,3,5,9,12]`, `target = 2`

Output: `-1`

Explanation: 2 does not exist in `nums` so return -1

Constraints:

- $1 \leq \text{nums.length} \leq 10^4$
- $-10^4 < \text{nums}[i], \text{target} < 10^4$
- All the integers in `nums` are **unique**.
- `nums` is sorted in ascending order.

For example:

Test	Result
<code>print(search([-1,0,3,5,9,12],9))</code>	4

Answer: (penalty regime: 0 %)

Reset answer

```

1 | def search(nums: list[int], target: int) -> int:
2 |     left, right = 0, len(nums) - 1
3 |     while left <= right:
4 |         mid = (left + right) // 2
5 |         if nums[mid] == target:
6 |             return mid
7 |         elif nums[mid] < target:
8 |             left = mid + 1
9 |         else:
10 |             right = mid - 1
11 |     return -1

```

	Test	Expected	Got	
✓	print(search([-1,0,3,5,9,12],9))	4	4	✓
✓	print(search([-1,0,3,5,9,12],2))	-1	-1	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Question 10 | Correct Mark 1.00 out of 1.00

Two string values S1, S2 are passed as the input. The program must print first N characters present in S1 which are also present in S2.

Input Format:

The first line contains S1.

The second line contains S2.

The third line contains N.

Output Format:

The first line contains the N characters present in S1 which are also present in S2.

Boundary Conditions:

$2 \leq N \leq 10$

$2 \leq \text{Length of S1, S2} \leq 1000$

Example Input/Output 1:

Input:

```
abcbde
cdefghbb
3
```

Output:

```
bcd
```

Note:

b occurs twice in common but must be printed only once.

Answer: (penalty regime: 0 %)

```
1 s1=input().strip()
2 s2=input().strip()
3 n=int(input())
4 s2_set=set(s2)
5 result=[]
6 used=set()
7 for char in s1:
8     if char in s2_set and char not in used:
9         result.append(char)
10        used.add(char)
11        if len(result)==n:
12            break
13 print(''.join(result))
```

```
13 | print( result );
```

	Input	Expected	Got	
✓	abcbde cdefghbb 3	bcd	bcd	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.