Question 1

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
a=input()
b=input()
c=""
for i in a :
    if i not in b:
        c+=i
print(c)
```

	Input	Expected	Got	

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

Answer: (penalty regime: 0 %)

Reset answer

```
def searchMatrix(m: list[list[int]], target: int) -> bool:
    for i in range(len(m)) :
        for j in range(len(m)) :
            if m[i][j]==target :
                return True
    return False
```

Constraints:

1 <= s.length <= 1000

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

s is a balanced string.

For example:

Test	Result
`print(BalancedStrings('RLRRLLRLRL'))`	4
`print(BalancedStrings('RLLLLRRRLR'))`	3

Answer: (penalty regime: 0 %)

Reset answer

```
def BalancedStrings(s,l=0,r=0,count=0):
    for i in s :
        if i=='L' :
            l+=1
        elif i=='R' :
            r+=1
        if l==r :
            count+=1
    return count
```

ATS-1

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 %)

```
a=int(input())
p=input()
b=list(map(int,p.split()))
count=0
c=int(input())
for i in range(len(b)) :
    for j in range(i+1,len(b)) :
        if (b[i]+b[j])==c :
            print("Yes")
            count=1
            break
    if count==1 :
        break
if count==0 :
    print("No")
```

Question **5**

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
a=list(map(int,input().split(',')))
b=int(input())
c=0
flag=0
d=len(a)
a.sort()
while c<d :
    p=(c+d)//2
    if a[p]==b :
        print("True")
        flag=1
        break
    elif b<a[p] :
        d=p
    else :
        c=p+1
if flag==0 :
    print("False")
```

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
```

For example:

Test	Result
print(missingNumber([3,0,1]))	2
print(missingNumber([0,1]))	2

Answer: (penalty regime: 0 %)

Reset answer

```
def missingNumber(n):
    count=0
    flag=0
    p=len(n)-1
    for i in range(p) :
        count+=1
        if count not in n :
            flag=1
        if flag==1 :
            break
    if flag==1 :
        return count
    else :
        return n[p]+1
```

Sample Output

```
10 6
```

For example:

Input	Result
4 12 3 6 8	12 8

Answer: (penalty regime: 0 %)

```
a=int(input())
b=list(map(int,input().split()))
c=[]
d=len(b)-1
if a>1 :
    if b[0]>b[1] :
        c.append(b[0])
    if b[d]>b[d-1] :
        c.append(b[d])
for i in range(1,d-1) :
    m=i-1
    n=i+1
    if b[i]>b[m] and b[i]>b[n] :
        c.append(b[i])
c.sort(reverse=True)
print(*c)
```

Marks for this submission: 1.00/1.00.

Question **8**

Correct

Mark 1.00 out of 1.00

Flag question

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 %)

```
a=input().split(' ')
c=""
for i in a :
    i=i.lower()
    if i!=i[::-1] :
        c+=i+" "
print(c)
```

Constraints:

- `1 <= nums.length <= 10^4`
- `-10^4 < nums[i], target < 10^4`
- All the integers in `nums` are **unique**.
- `nums` is sorted in ascending order.

For example:

Test	Result
`print(search([-1,0,3,5,9,12],9))`	4

Answer: (penalty regime: 0 %)

Reset answer

```
def search(n: list[int], t: int) -> int:
    count=0
    flag=0
    for i in range(len(n)) :
        if n[i]==t :
            count=i
            flag=1
            break
    if flag==1 :
        return count
    else :
        return -1
```

abcbde
cdefghbb
3

Output:

bcd

Note:

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

Answer: (penalty regime: 0 %)

```
a=input()
b=input()
c=int(input())
d=""
count=0
for i in a :
    if count>=c :
        break
    if i in b and i not in d :
        d+=i
        count+=1
print(d)
```