Time taken	1 hour 27 mins
Marks	5.00/5.00
Grade	**100.00** out of 100.00

Output: 2

Explanation:

Pairs with sum K(= 13) are {(5, 8), (6, 7), (6, 7)}.

Therefore, distinct pairs with sum K(= 13) are { (5, 8), (6, 7) }.

Therefore, the required output is 2.

For example:

Input	Result
1,2,1,2,5 3	1
1,2 0	0

Answer: (penalty regime: 0 %)

```
n=input()
k=int(input())
lst=()
for i in str(n):
    if i != ",":
        lst+=(i,)
tup=lst


seen = set()
pairs = set()

for number in tup:
    for j in range(1,len(tup)):
        if k== int(number)+ int(tup[j]):

            seen.add(number)
            seen.add(tup[j])


print(int(len(seen))//2)

```

	Input	Expected	Got	
✓	5,6,5,7,7,8 13	2	2	✓
✓	1,2,1,2,5 3	1	1	✓

Given a string `s` that represents a **DNA sequence**, return all the **`10`-letter-long** sequences (substrings) that occur more than once in a DNA molecule. You may return the answer in **any order**.

Example 1:

```
Input: s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"
Output: ["AAAAACCCCC","CCCCCAAAAA"]
```

Example 2:

```
Input: s = "AAAAAAAAAAAAA"
Output: ["AAAAAAAAAA"]
```

For example:

Input	Result
AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT	AAAAACCCCC CCCCCAAAAA

Answer: (penalty regime: 0 %)

```
def Sequences(s):
    if len(s) < 10:
        return []

    count = {}
    result = []

    for i in range(len(s) - 9):
        sequence = s[i:i+10]
        if sequence in count:
            count[sequence] += 1
        else:
            count[sequence] = 1

    for sequence, c in count.items():
        if c > 1:
            result.append(sequence)

    return result


s = input()
result = Sequences(s)



for sequence in result:
    print(sequence)
```

Marks for this submission: 1.00/1.00.

```
Input: nums = [1,3,4,2,2]
Output: 2
```

Example 2:

```
Input: nums = [3,1,3,4,2]
Output: 3
```

For example:

Input	Result
1 3 4 4 2	4

Answer: (penalty regime: 0 %)

```
a=[]
b = input()
a.append(b)
b = str(a)
b.split()
c=[]
d = []
for i in b:
    if i not in c:
        if chr(48)<i<chr(57):
            c.append(i)
    elif i in c:
        if chr(48)<i<chr(57):
            d.append(i)
print("".join(d))
```

	Input	Expected	Got	
✓	1 3 4 4 2	4	4	✓
✓	1 2 2 3 4 5 6 7	2	2	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

The first line contains space-separated values, denoting the size of the two arrays in integer format respectively.

The next two lines contain the space-separated integer arrays to be compared.

Sample Input:

5 4

1 2 8 6 5

2 6 8 10

Sample Output:

1 5 10

3

Sample Input:

5 5

1 2 3 4 5

1 2 3 4 5

Sample Output:

NO SUCH ELEMENTS

For example:

Input	Result
5 4 1 2 8 6 5 2 6 8 10	1 5 10 3

Answer: (penalty regime: 0 %)

```
def main():
    sizes = input().strip().split()
    size1, size2 = int(sizes[0]), int(sizes[1])

    array1 = list(map(int, input().strip().split()))
    array2 = list(map(int, input().strip().split()))

    set1 = set(array1)
    set2 = set(array2)

    non_repeating_elements = set1.symmetric_difference(set2)

    if not non_repeating_elements:
        print("NO SUCH ELEMENTS")
    else:
        print(" ".join(map(str, sorted(non_repeating_elements))))
        print(len(non_repeating_elements))

if __name__ == "__main__":
    main()
```

	10 11 12			

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.

Output: Yes

Input: str = "REC101"

Output: No

For example:

Input	Result
01010101010	Yes
010101 10101	No

Answer: (penalty regime: 0 %)

```
n=str(input())
l=[]
for i in n:
    if i=="0" or i=="1":

        l.append(i)

if len(l)==len(n):
    print("Yes")
else:
    print("No")
```

	Input	Expected	Got	
✓	01010101010	Yes	Yes	✓
✓	REC123	No	No	✓
✓	010101 10101	No	No	✓

Passed all tests! ✓

Correct

Marks for this submission: 1.00/1.00.