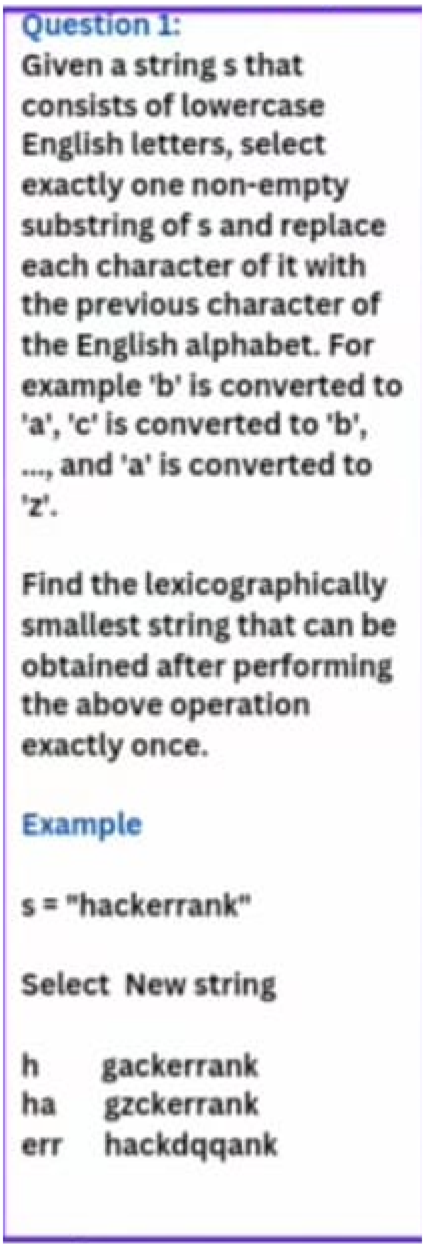
Question:



Question 1:

Given a string s that consists of lowercase English letters, select exactly one non-empty substring of s and replace each character of it with the previous character of the English alphabet. For example 'b' is converted to 'a', 'c' is converted to 'b', ..., and 'a' is converted to 'z'.

Find the lexicographically smallest string that can be obtained after performing the above operation exactly once.

Example

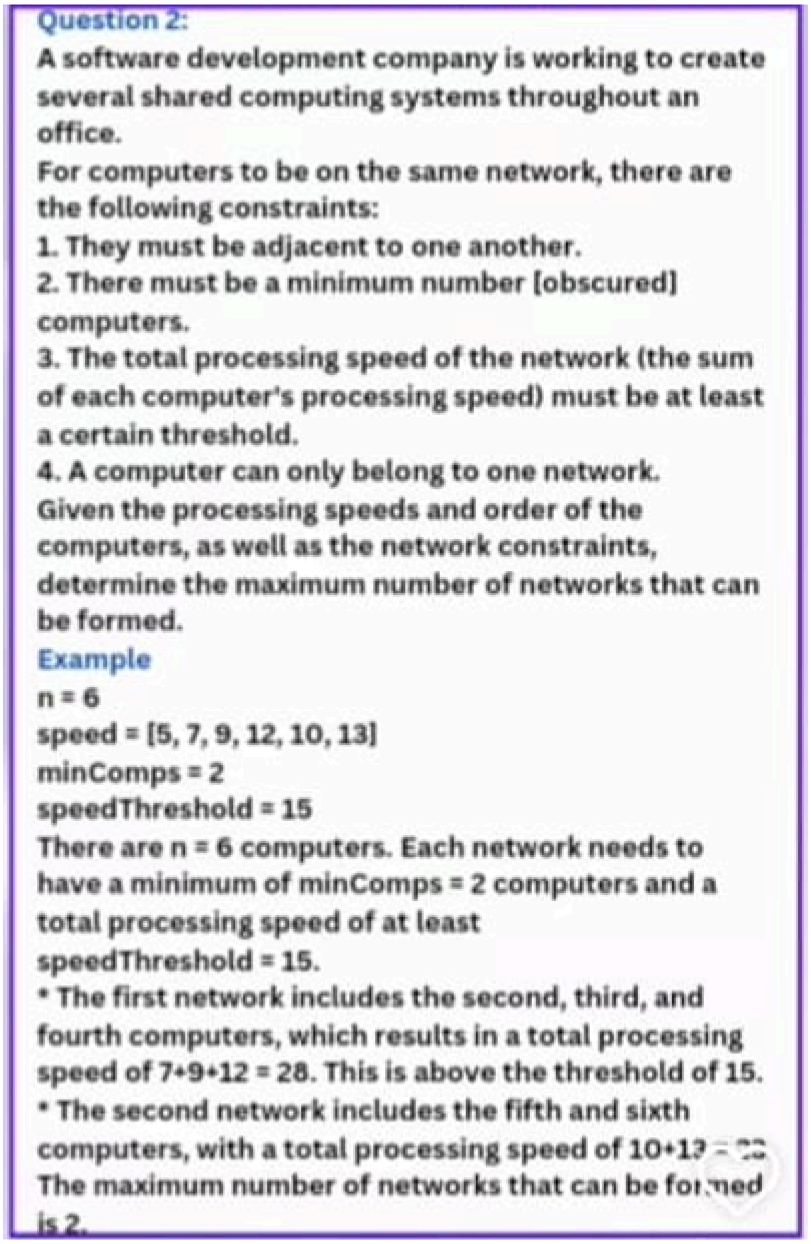
s = "hackerrank"

Select	New string
h	gackerrank
ha	gzckerrank
err	hackdqqank

Code:

```
s = "hackerrank"
n = len(s)
new_string = ""
for i in range(n):
    if s[i] == 'a':
        new_string += 'z'
    else:
        new_string += chr(ord(s[i]) - 1)
lexico_smallest = min(s, new_string)
print(lexico_smallest)
```

Question-2:



Question 2:

A software development company is working to create several shared computing systems throughout an office.

For computers to be on the same network, there are the following constraints:

1. They must be adjacent to one another.
2. There must be a minimum number [obscured] computers.
3. The total processing speed of the network (the sum of each computer's processing speed) must be at least a certain threshold.
4. A computer can only belong to one network.

Given the processing speeds and order of the computers, as well as the network constraints, determine the maximum number of networks that can be formed.

Example

n = 6
speed = [5, 7, 9, 12, 10, 13]
minComps = 2
speedThreshold = 15

There are n = 6 computers. Each network needs to have a minimum of minComps = 2 computers and a total processing speed of at least speedThreshold = 15.

* The first network includes the second, third, and fourth computers, which results in a total processing speed of 7+9+12 = 28. This is above the threshold of 15.
* The second network includes the fifth and sixth computers, with a total processing speed of 10+13 = 23

The maximum number of networks that can be formed is 2.

Code:

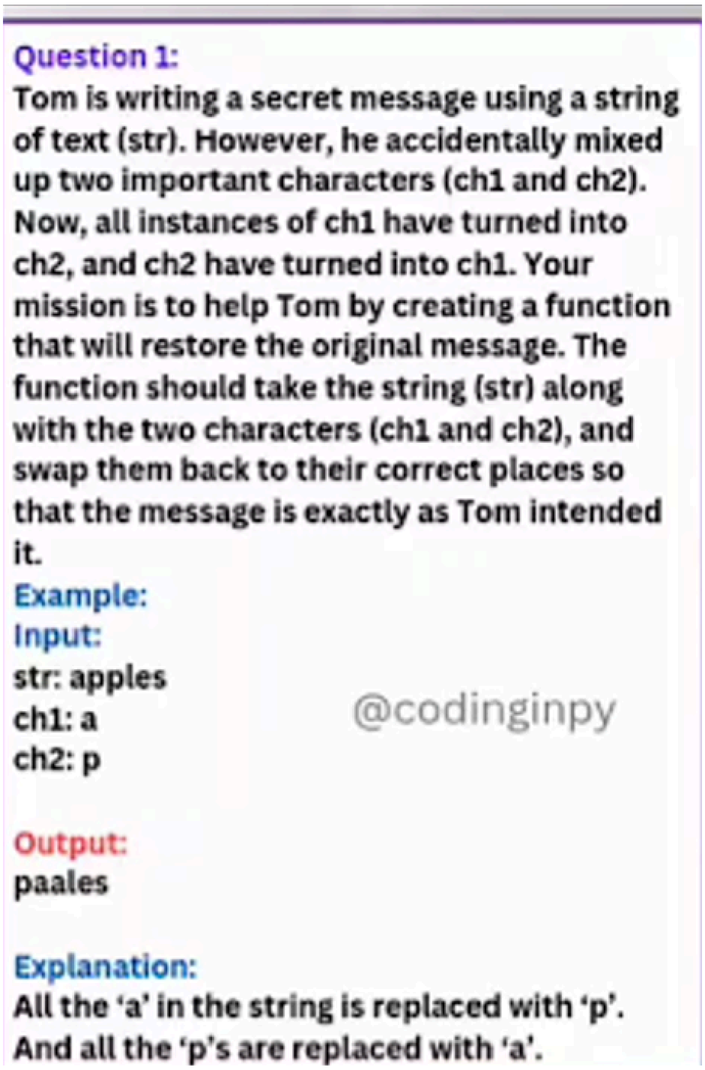
```
n = 6
speed = [5, 7, 9, 12, 10, 13]
min_comps = 2
speed_threshold = 15
speed.sort(reverse=True)
total_networks = 0
i = 0
```

```
while i < n:
    total_speed = 0
    count_comps = 0
    while i < n and count_comps < min_comps:
        total_speed += speed[i]
        count_comps += 1
        i += 1

    if total_speed >= speed_threshold:
        total_networks += 1
    else:
        break

print(total_networks)
```

Question-3:



Question 1:

Tom is writing a secret message using a string of text (str). However, he accidentally mixed up two important characters (ch1 and ch2). Now, all instances of ch1 have turned into ch2, and ch2 have turned into ch1. Your mission is to help Tom by creating a function that will restore the original message. The function should take the string (str) along with the two characters (ch1 and ch2), and swap them back to their correct places so that the message is exactly as Tom intended it.

Example:

Input:

str: apples
ch1: a
ch2: p

Output:

paales

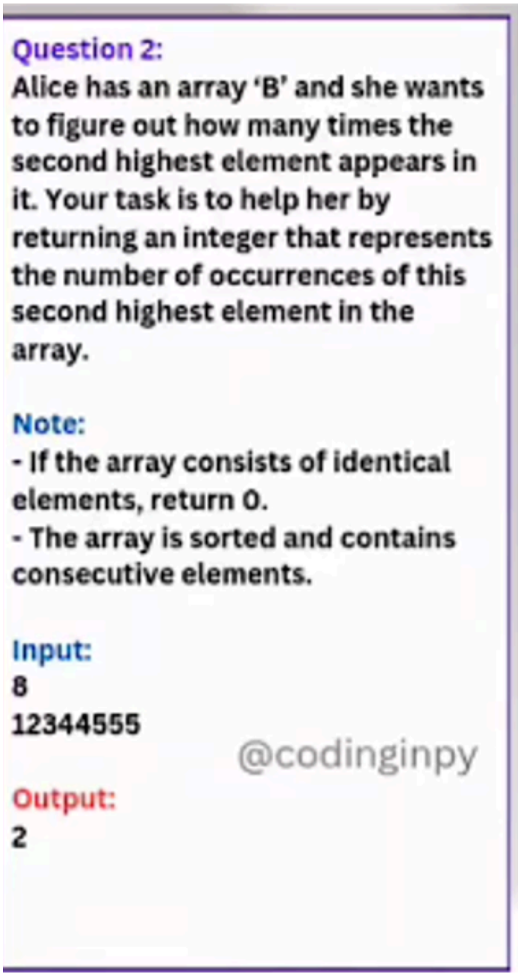
Explanation:

All the 'a' in the string is replaced with 'p'. And all the 'p's are replaced with 'a'.

Code:

```
s = input("Enter the string: ")
ch1 = input("Enter first character to swap: ")
ch2 = input("Enter second character to swap: ")
res = ''
for i in s:
    if i == ch1:
        res += ch2
    elif i == ch2:
        res += ch1
    else:
        res += i
print(res)
```

Question-4:



Question 2:
Alice has an array 'B' and she wants to figure out how many times the second highest element appears in it. Your task is to help her by returning an integer that represents the number of occurrences of this second highest element in the array.

Note:
- If the array consists of identical elements, return 0.
- The array is sorted and contains consecutive elements.

Input:

```
8
12344555
```

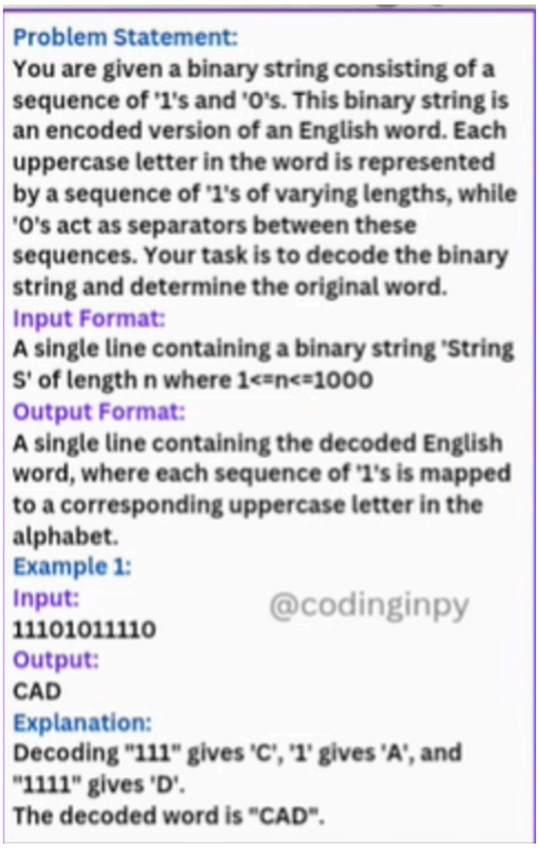
Output:

```
2
```

Code:

```
n = int(input())
B = list(map(int, input().split()))
second_highest = None
highest = None
for num in B:
    if highest is None or num > highest:
        second_highest = highest
        highest = num
    elif second_highest is None or num > second_highest and num != highest:
        second_highest = num
count = 0
for num in B:
    if num == second_highest:
        count += 1
print(count)
```

Question-5:



Problem Statement:

You are given a binary string consisting of a sequence of '1's and '0's. This binary string is an encoded version of an English word. Each uppercase letter in the word is represented by a sequence of '1's of varying lengths, while '0's act as separators between these sequences. Your task is to decode the binary string and determine the original word.

Input Format:

A single line containing a binary string 'String S' of length n where 1<=n<=1000

Output Format:

A single line containing the decoded English word, where each sequence of '1's is mapped to a corresponding uppercase letter in the alphabet.

Example 1:

Input:

11101011110

Output:

CAD

Explanation:

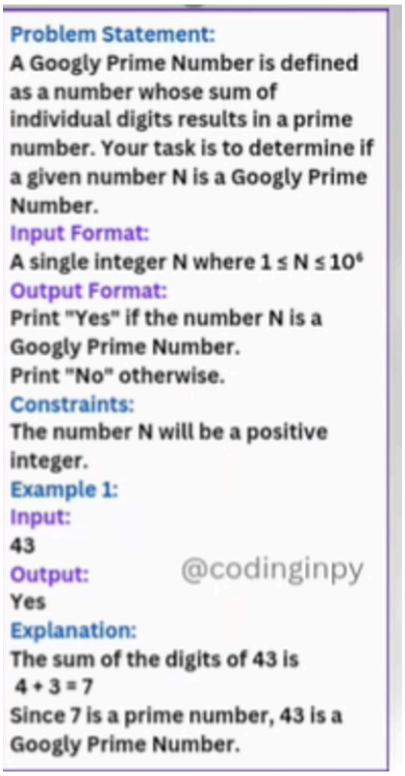
Decoding "111" gives 'C', '1' gives 'A', and "1111" gives 'D'.

The decoded word is "CAD".

Code:

```
hash_map = {i: chr(64 + i) for i in range(1, 27)}
S = input()
decoded_word = ""
count = 0
for char in S:
    if char == '1':
        count += 1
    elif char == '0':
            decoded_word += hash_map[count]
            count = 0
print(decoded_word)
```

Question-6:



Problem Statement:
A Googly Prime Number is defined as a number whose sum of individual digits results in a prime number. Your task is to determine if a given number N is a Googly Prime Number.

Input Format:
A single integer N where $1 \le N \le 10^6$

Output Format:
Print "Yes" if the number N is a Googly Prime Number.
Print "No" otherwise.

Constraints:
The number N will be a positive integer.

Example 1:

Input:
43

Output:
Yes

Explanation:
The sum of the digits of 43 is
4 + 3 = 7
Since 7 is a prime number, 43 is a Googly Prime Number.

Code:

```
n=int(input())
t=n
sum=0
while n!=0:
    sum=sum+n%10
    n=n//10
f=1
c=0
while (f<=sum):
    if (sum%f==0):
        c=c+1
    f=f+1
if(c==2):
    print(t, " is a googly prime number")
```

Question-7:

Remove brackets from an algebraic expression.

Code:

```
s=input()
str=''
for i in s:
    if((ord(i)!=40 and (ord(i)!=41 and (ord(i)!=91 and ord(i)!=93 and (ord(i)!=123 and
(ord(i)!=125)))))):
        str+=i
print(str)
```