**LONGEST PEAK SUBARRAY**

def longest\_peak\_subarray(arr):

n = len(arr)

if n < 3:

return 0

max\_len = 0

i = 1

while i < n - 1:

if arr[i - 1] < arr[i] > arr[i + 1]:

left = i - 1

right = i + 1

while left > 0 and arr[left - 1] < arr[left]:

left -= 1

while right < n - 1 and arr[right] > arr[right + 1]:

right += 1

peak = arr[i]

is\_unique\_peak = True

for j in range(left, right + 1):

if j != i and arr[j] == peak:

is\_unique\_peak = False

break

if is\_unique\_peak:

max\_len = max(max\_len, right - left + 1)

i = right

else:

i += 1

return max\_len

n = int(input())

arr = list(map(int, input().split()))

print(longest\_peak\_subarray(arr))

**THE FORGOTTEN SONG 1**

def longest\_repeating\_substring(s):

def is\_valid(mid):

seen = set()

for i in range(len(s) - mid + 1):

substring = s[i:i + mid]

if substring in seen:

return substring

seen.add(substring)

return None

left, right = 0, len(s)

result = "-1"

while left <= right:

mid = (left + right) // 2

found = is\_valid(mid)

if found:

result = found

left = mid + 1

else:

right = mid - 1

return result

s = input().strip()

print(longest\_repeating\_substring(s))

**THE LOST TREASURE TRAIL**

def max\_subarray\_sum(arr):

max\_sum = float('-inf')

current\_sum = 0

max\_length = 0

current\_length = 0

for num in arr:

current\_sum += num

current\_length += 1

if current\_sum > max\_sum:

max\_sum = current\_sum

max\_length = current\_length

if current\_sum < 0:

current\_sum = 0

current\_length = 0

return max\_sum

n=int(input())

arr=list(map(int,input().split()))

print(max\_subarray\_sum(arr))

**THE SECRET CODE**

def min\_window(S, T):

def contains(window, T):

for char in T:

if char not in window or window.count(char) < T.count(char):

return False

return True

min\_len = float('inf')

min\_window = ""

for i in range(len(S)):

for j in range(i + 1, len(S) + 1):

window = S[i:j]

if contains(window, T) and len(window) < min\_len:

min\_len = len(window)

min\_window = window

return min\_window

S = input()

T = input()

print( min\_window(S, T))

**THE JUMBLED INVITATIONS**

def are\_anagrams(str1, str2):

str1 = str1.replace(" ", "").lower()

str2 = str2.replace(" ", "").lower()

return sorted(str1) == sorted(str2)

str1 = input()

str2 = input()

if are\_anagrams(str1, str2):

print("True")

else:

print("False")