

question1

""There is a ant on your balcony. It wants to leave the rail so sometimes it moves right

and sometimes it moves left until it gets exhausted. Given an integer array A of size N which consists of integer 1 and -1 only representing ant's moves. Where 1 means ant moved unit distance towards the right side and -1 means it moved unit distance towards the left. Your task is to find and return the integer value representing how many times the ant reaches back to original starting position. Note:

- Assume 1-based indexing
- Assume that the railing extends infinitely on the either sides Input Format: input1 : An integer value N representing the number of moves made by the ant. input2 : An integer array A consisting of the ant's moves towards either side"

In [1]:

```

1 n=int(input())
2 arr=list(map(int,input().split()))
3 count=0
4 for i in range(n):
5     if(sum(arr[:i+1])==0):
6         count+=1
7 print(count)

```

```

5
-1 -1 -1
0

```

question2

You are given an integer array of size N, representing jars of chocolates. Three

students A, B, and C respectively, will pick chocolates one by one from each chocolate jar, till the jar is empty, and then repeat the same with the rest of the jars. Your task is to fine and return an integer value representing the total number of chocolates that student A will have, after all the chocolates have been picked from all the jars. Note: Once a jar is done A will start taking the chocolates from the new jar. Input Format : input1: An integer array representing the quantity of chocolates in each jar. input2: An integer value N representing the number of jars. Output Format: Return an integer value representing the total number of chocolates that student A will have, after all the chocolates are picked

In [12]:

```

1 arr=list(map(int,input().split()))
2 n=int(input())
3 c=0
4 for i in arr:
5     if i==0:
6         continue
7     if(i<=3):
8         c=c+1
9     else:
10        if(i%3==0):
11            c=c+i//3
12        else:
13            c=c+(i//3)+1
14 print(c)

```

```

10 20 30
3
21

```

question3

Max is having a dog . he want to find the age of the dog with respect to the age of human.

he came to know that , the age of the dog is mesured with respect to human has a formula to proceed. example: 1 year of life span of dog is same as seveen years of life span of human being Now , calculate the age of MAX dog.

In [4]:

```

1
2 def calculate_age(n):
3     return n*7
4 n=int(input())
5 print(calculate_age(n))
6

```

```

6
42

```

question4

```

1 ### Max is planning to take part in a Diwali contest at a Diwali Party that will begin at 8
2 PM and will run until midnight (12 AM) i.e., for 4 hours. He also needs
3 to travel to the
4 party venue within this time which takes him P minutes. The contest
5 comprises
6 of N problems that are arranged in order of difficulty, with problem 1
7 being the
8 simplest and problem N being the most difficult. Max is aware that he
9 will require  $5*i$ 

```

6 minutes to solve the i th problem.
 7 Your task is help Max find and return an integer value, representing the number of
 8 problems Max can solve and reach the party venue within the given time frame of 4
 9 hours.
 10 Note: Max will leave his home at exactly 8 PM to reach the party venue.
 11 Input Format:
 12 input1: An integer value N , representing the total number of problems.
 13 input2: An integer value P , Representing the time to travel in minutes from his home
 14 to the party venue.

In [5]:

```
1 inp1=int(input())
2 inp2=int(input())
3 problem_solved=0
4 remainng_time=240-inp2
5 for i in range(1,inp1+1):
6     if(remainng_time>0 and remainng_time>5*i):
7         remainng_time=remainng_time-5*i
8         problem_solved+=1
9 print(problem_solved)
```

```
240
60
7
```

question 5

1 **### Paul is given an array A of length N. He must perform the following Operations on the array sequentially:**
 2 * Choose any two integers from the array and calculate their average.
 3 * If an element is less than the average, update it to 0. However, if the element is
 4 greater than or equal to the average, he need not update it.
 5 Your task is to help Paul find and return an integer value, representing the minimum
 6 possible sum of all the elements in the array by performing the above operations. Note: An exact average should be calculated, even if it results in a decimal.
 7 Input Format:
 8 input1: An integer value N , representing the size of the array A.
 9 input2: An integer array A.
 10 Output Format:
 11 Return an integer value, representing the minimum possible sum of all the elements
 12 in the array by
 13 Sample Input
 14 5
 15 1 2 3 4 5
 16 Sample Output

In [1]: 18 | 5

```

1 n=int(input())
2 arr=list(map(int,input()))
3 arr.sort()
4 element_1=arr[-1]
5 element_2=arr[-2]
6 avg=(element_1+element_2)/2
7 sum=0
8 for i in range(len(arr)):
9     if(arr[i]>=avg):
10         sum=sum+arr[i]
11 print(sum)
12

```

240
60
6

question6

Minimum Number of Key Presses

George has a setup which includes a special keyboard and a monitor , that initially displays 0. The special keyboard has 11 numeric keys (0,1,2,3,4,5,6,7,8,9,00). If he presses 00, the previously displayed value will be multiplied by 100. Whereas, if he presses any other numeric key, the previously displayed value will be firstly multiplied by 10 and then the number on the key will be added to it You are given a numeric string S. Your task is to help George find and return an integer value, representing the minimum number of key presses to reach the number.

Input Specification: input: A numeric string s. representing the final number,

Output Specification: Return an integer value, representing the minimum number of key presses to reach the number. **Sample Input:** 100 **Sample Output:** 2

In []: 1

question7

Special String

Alice has a string A consisting of lowercase English letters. Her friend gives her another string S and asks her to modify string A and replace its characters with the characters present in string S. But, to achieve the above task, Alice must follow the below steps:

1. Choose a character from string S that has the minimum ASCII distance from the ith character in string A Replace the ith character in string A with the chosen character in string S Your task is to find and return an integer value, representing minimum total ASCII distance that is required to modify string A to the characters in string S. Return 0, if all the

characters in string S are already present in string A Sample Input: abcd xyz Sample Output: 86

question8

The function accepts two positive integers ‘r’ and ‘unit’ and a positive integer array ‘arr’ of size ‘n’ as its argument ‘r’ represents the number of rats present in an area, ‘unit’ is the amount of food each rat consumes and each ith element of array ‘arr’ represents the amount of food present in ‘i+1’ house number, where $0 \leq i < n$

Note:

Return -1 if the array is null Return 0 if the total amount of food from all houses is not sufficient for all the rats. Computed values lie within the integer range. Example:

Input:

r: 7 unit: 2 n: 8 arr: 2 8 3 5 7 4 1 2 Output:

4

Explanation: Total amount of food required for all rats = $r * \text{unit}$

$$= 7 * 2 = 14.$$

The amount of food in 1st houses = $2+8+3+5 = 18$. Since, amount of food in 1st 4 houses is sufficient for all the rats. Thus, output is 4.

```
In [12]: 1 def sufficient_food(r, unit, arr):
2     if arr is None:
3         return -1
4
5     total_food_required = r * unit
6     total_food_available = 0
7     houses_with_food = 0
8
9     for food_in_house in arr:
10        total_food_available += food_in_house
11        houses_with_food += 1
12        if total_food_available >= total_food_required:
13            return houses_with_food
14
15    return 0
16
17 r = 7
18 unit = 2
19 arr = [2,8,3,5,7,4,1,2]
20 print(sufficient_food(r, unit, arr))
```

4

question9

The Binary number system only uses two digits, 0 and 1 and number system can be called binary string. You are required to implement the following function:

```
int OperationsBinaryString(char* str);
```

The function accepts a string str as its argument. The string str consists of binary digits separated with an alphabet as follows:

– A denotes AND operation – B denotes OR operation – C denotes XOR Operation You are required to calculate the result of the string str, scanning the string to right taking one operation at a time, and return the same.

Note:

No order of priorities of operations is required Length of str is odd If str is NULL or None (in case of Python), return -1 Input: str: 1C0C1C1A0B1

Output: 1

Explanation: The alphabets in str when expanded becomes “1 XOR 0 XOR 1 XOR 1 AND 0 OR 1”, result of the expression becomes 1, hence 1 is returned.

Sample Input: 0C1A1B1C1C1B0A0

Output: 0

```
In [18]: 1 def OperationsBinaryString(string):
2     if string is None:
3         return -1
4
5     result = int(string[0])
6
7     for i in range(1, len(string), 2):
8         operation = string[i]
9         num = int(string[i+1])
10
11         if operation == 'A':
12             result &= num
13         elif operation == 'B':
14             result |= num
15         elif operation == 'C':
16             result ^= num
17         else:
18             return -1
19     return result
20
21
22 print(OperationsBinaryString("1C0C1C1A0B1"))
23 print(OperationsBinaryString(" 0C1A1B1C1C1B0A0"))
24
```

```
1
0
```

```
In [ ]: 1 def OperationsBinaryString(str):
2     a=int(str[0])
3     i=1
4     while i< len(str):
5         if str[i]=='A':
6             a=a&int(str[i+1])
7         elif str[i]=='B':
8             a|=int(str[i+1])
9         else:
10            a=a^int(str[i+1])
11         i+=2
12     return a
13 str=input()
14 print(OperationsBinaryString(str))
15
```

question10

You are given a function.

int CheckPassword(char str[], int n); The function accepts string str of size n as an argument. Implement the function which returns 1 if given string str is valid password else 0. str is a valid password if it satisfies the below conditions.

- At least 4 characters – At least one numeric digit – At Least one Capital Letter – Must not have space or slash (/) – Starting character must not be a number Assumption: Input string will not be empty.

Example:

Input 1: aA1_67 Input 2: a987 abC012

Output 1: 1 Output 2: 0

```
In [2]: 1 def CheckPassword(s,n):
2     if n<4:
3         return 0
4     if s[0].isdigit():
5         return 0
6     cap=0
7     nu=0
8     for i in range(n):
9         if s[i]==' ' or s[i]=='/':
10             return 0
11         if s[i]>='A' and s[i]<='Z':
12             cap+=1
13         elif s[i].isdigit():
14             nu+=1
15     if cap>0 and nu>0:
16         return 1
17     else:
18         return 0
19
20 s=input()
21 a=len(s)
22 print(CheckPassword(s,a))
```

aA1_67

1

question11

'''You are given a function,

```
int findCount(int arr[], int length, int num, int diff);
```

The function accepts an integer array ‘arr’, its length and two integer variables ‘num’ and ‘diff’. Implement this function to find and return the number of elements of ‘arr’ having an absolute difference of less than or equal to ‘diff’ with ‘num’. Note: In case there is no element in ‘arr’ whose absolute difference with ‘num’ is less than or equal to ‘diff’, return -1.

Example: Input:

arr: 12 3 14 56 77 13 num: 13 diff: 2 Output: 3

Explanation: Elements of ‘arr’ having absolute difference of less than or equal to ‘diff’ i.e. 2 with ‘num’ i.e. 13 are 12, 13 and 14.”

```
In [1]: 1 def findCount(n, arr, num, diff):
2     count=0
3     for i in range(n):
4         if(abs(arr[i]-num)<=diff):
5             count+=1
6     if count:
7         return count
8     return 0
9 n=int(input())
10 arr=list(map(int,input().split()))
11 num=int(input())
12 diff=int(input())
13 print(findCount(n, arr, num, diff))
```

12 3 14 56 77 13

ValueError

Traceback (most recent call last)

```
Cell In[1], line 9
      7     return count
      8     return 0
----> 9 n=int(input())
     10 arr=list(map(int,input().split()))
     11 num=int(input())
```

ValueError: invalid literal for int() with base 10: '12 3 14 56 77 13'

question12

'''The function accepts two integers n, m as arguments Find the sum of all numbers in range from 1 to m(both inclusive) that are not divisible by n. Return difference between sum of integers not divisible by n with sum of numbers divisible by n.

Assumption:

n>0 and m>0 Sum lies between integral range Example

Input n:4 m:20 Output 90'''

```
In [4]: 1 def sum_difference(n, m):
2
3     total_sum = m * (m + 1) // 2
4
5     k = m // n
6     divisible_sum = n * k * (k + 1) // 2
7
8
9     not_divisible_sum = total_sum - divisible_sum
10
11
12     return not_divisible_sum - divisible_sum
13
14
15 n = 4
16 m = 20
17 print(sum_difference(n, m))
```

90

question13

Toss and score You are playing a game of Toss and Score in the Hillwood City Mall with your friends. The game consists of the following rules: Toss an unbiased coin multiple times. For each heads you get 2 points and for each tails you lose 1 point. The game ends as soon as you get 3 heads in a row, or you toss the coin throughout the length of string S. You have been given a string S consisting of letters H (for heads) and T (for tails) denoting the sequence results you get on the toss of coin N times. Your task is to find and return an integer value representing the final score you get once the game ends. Note: The final score can be negative too. Input Specification: Input1: A string s. representing the sequence of results you get on the toss of coin N times Sample Input: HHHTT Output: 6 ""

```
In [11]: 1 def toss_and_score(s: str) -> int:
2     score = 0
3     consecutive_heads = 0
4
5     for char in s:
6         if char == 'H':
7             score += 2
8             consecutive_heads += 1
9             if consecutive_heads == 3:
10                 break
11         elif char == 'T':
12             score -= 1
13             consecutive_heads = 0
14
15     return score
16
17 sample_input = "HTTHHTHHHTHTH"
18 print(toss_and_score(sample_input))
```

7

```
In [20]: 1 toss=input()
2 head=0
3 score=0
4 for i in toss:
5     if(i=="H"):
6         head+=1
7         score+=2
8         if(head==3):
9             break
10    else:
11        score-=1
12        head=0
13 print(score)
```

HHHTT

6

question14

19. Nearest Corner Bruce is a newly hired employee at a company. The Office Management Department has given him a desk number, which is stored in string S. He has also been handed a string array A. containing all the N office desk numbers. Array A also includes the symbol"-", which stands for the gap in the sitting arrangement. Comer seats are those that are on either side of the gap. Your task is to help Bruce find and return an integer value. representing how far he is from the nearest corner seat. Return 0, if he is in the corner seat. Note: There will always be at least one gap in the string array A Desk number is always in a format of a number first followed by an English letter in uppercase Assume 0 - based indexing Input Specification: A string S. representing Bruce's newly assigned desk

number. Second line containing space seperated strings showing the seat positions and gaps Sample input: 3C 1A 2B - 3C 4D Sample Output: 0 ""

```
def nearest_corner(S, A):
    seats = A.split()

    bruce_index = seats.index(S)

    gap_indices = [i for i, seat in enumerate(seats) if seat == "-"]

    min_distance = float('inf')

    for gap in gap_indices:
        if gap > 0:
            min_distance = min(min_distance, abs(gap - 1 - bruce_index))

        if gap < len(seats) - 1:
            min_distance = min(min_distance, abs(gap + 1 - bruce_index))

    return min_distance

S = "3C" A = "1A 2B - 3C 4D" print(nearest_corner(S, A))
```

In [27]:

```
1 def nearest_corner(S, A):
2
3     seats = A.split()
4
5     bruce_index = seats.index(S)
6
7     gap_indices = [i for i, seat in enumerate(seats) if seat == "-"]
8
9     min_distance = float('inf')
10
11    for gap in gap_indices:
12        if gap > 0:
13            min_distance = min(min_distance, abs(gap - 1 - bruce_index))
14
15        if gap < len(seats) - 1:
16            min_distance = min(min_distance, abs(gap + 1 - bruce_index))
17
18    return min_distance
19
20 S = "3C"
21 A = "1A 2B - 3C 4D"
22 print(nearest_corner(S, A))
```

0

question15

30. Boring Arrays You are given an array A of size N. In one operation you can select any two elements from it, add their absolute difference in your score. Your task is to find and return an integer value, representing the maximum score. Note: Assume 1 based indexing The elements on which operation has been performed cannot be selected again. Input Specification: Input1: An integer value N, representing the size of array A input2: An integer array A Output Specification: Return an integer value, representing the maximum score Sample Input: 4 1 2 3 4 Sample Output: 4 "

In [22]:

```
1 def max_score(N, A):
2     A.sort()
3
4     score = 0
5
6     left = 0
7     right = N - 1
8
9     while left < right:
10         score += abs(A[right] - A[left])
11         left += 1
12         right -= 1
13
14     return score
15
16 N = 4
17 A = [1, 2, 3, 4]
18 print(max_score(N, A))
```

4

```
In [29]: 1 def max_score(N, A):
2
3     A.sort()
4
5     score = 0
6
7     while len(A) > 1:
8
9         left = A.pop(0)
10        right = A.pop(-1)
11
12        score += abs(right - left)
13    return score
14 N = 4
15 A = [1, 2, 3, 4]
16 print(max_score(N, A))
17
```

4

```
In [36]: 1 n=int(input())
2 a=list(map(int,input().split()))
3 start=0
4 end=-1
5 res=[]
6 while(len(a)>1):
7     res.append(abs(a[start]-a[end]))
8     a.pop(start)
9     a.pop(end)
10    print(max(res))
```

4

1 2 3 4

3

question16

Problem Statement: In a quaint village nestled between rolling hills, there were N different salt containers and N different pepper containers in two separate groups. Each container had a specific level of bitterness, represented by arrays A and B respectively. The task at hand was to form N combinations, each consisting of one salt container and one pepper container. However, there was a twist to the challenge. The objective was to arrange the combinations in such a way that the maximum bitterness level, which is the sum of salt and pepper quantities in each combination, was minimized. Print the lowest possible maximum bitterness level.

Input Format: The first line contains a single integer N, the number of salt and pepper containers in each group. The second line contains N space-separated integers, denoting the bitterness level of N salt containers. The third line contains N space-separated integers, denoting the bitterness level of N pepper containers.

Sample Input: 3 1 3 5 2 8 6

Sample Output: 11

"

```
In [1]: 1 def solver(n,salt,pepper):
2     #salt=0
3     #pepper=0
4     r=[]
5     for i in range(len(salt)):
6         r.append(salt[i]+pepper[i])
7     return max(r)
8 n=int(input())
9 salt=list(map(int,input().split()))
10 pepper=list(map(int,input().split()))
11 res=solver(n,salt,pepper)
12 print(res)
```

```
3
1 3 5
2 8 6
11
```

question17

"

Arduino Tom is an Arduino Programmer. He has designed a program to run his robocar on a horizontal number line. Initially, the car is parked at: 0. Given an array A of N integers which can be A. B. C... the robocar runs as follows as per the designed program First the robocar moves A units in specified direction(right in case the integer is positive and left if the integer is negative). Then robocar first moves A units and then B units in a specified direction. In the next step, the robocar moves A units. B units, and then C units in a specified direction. This process keeps on repeating as per the number of integers in the sequence.. Your task is to find and return an integer value, representing the farthest coordinate reached by the robocar from the beginning to the end of the process. Sample Input: 1 -2 3 4 Sample Output: 6 "

```
In [5]: 1 def farthest_coordinate(arr):
2     max_distance = 0
3     current_position = 0
4     cumulative_movement = 0
5
6     for i in range(len(arr)):
7         cumulative_movement += arr[i]
8         current_position += cumulative_movement
9         max_distance = max(max_distance, abs(current_position))
10
11    return max_distance
12
13 # Sample Input
14 arr = [1, -2, 3, 4]
15 # Sample Output
16 print(farthest_coordinate(arr))
```

8

question18**""Pizza Party**

Angela has decided to throw a pizza party. she has ordered N number of pizzas to be served to her N number of friends. In this way, she will be serving only one pizza to each friend. She now wants to invite fewer people to her party in order to provide more pizzas per person. But at the same time, she wants to ensure that there are at least Y friends at her party. Your task is to help Angela find and return an integer value, representing the sum of digits of the minimum number of friends that she can invite to the party, ensuring that each person gets an equal number of pizzas Sample Input: 100 17 Sample Output:2"

```
In [6]: 1 def sum_of_digits(n):
2     return sum(int(digit) for digit in str(n))
3
4 def min_friends_sum_of_digits(N, Y):
5     for X in range(Y, N + 1):
6         if N % X == 0:
7             return sum_of_digits(X)
8     return -1
9 N = 200
10 Y = 19
11
12 print(min_friends_sum_of_digits(N, Y))
```

2

question24**"happy fathers day"**

extract the vowel which has max count.

```
In [1]: 1 def max_vowel_count(phrase):
2     vowels = "aeiou"
3
4     vowel_count = {vowel: 0 for vowel in vowels}
5
6     phrase = phrase.lower()
7
8     for char in phrase:
9         if char in vowels:
10            vowel_count[char] += 1
11
12     max_vowel = max(vowel_count, key=vowel_count.get)
13
14     return max_vowel, vowel_count[max_vowel]
15
16 phrase = "happy fathers day"
17 max_vowel, count = max_vowel_count(phrase)
18 print(f"The vowel '{max_vowel}' has the maximum count of {count} in the ph
```

The vowel 'a' has the maximum count of 3 in the phrase 'happy fathers day'.

```
In [7]: 1 from collections import Counter
2 def countofvowels(str):
3     for i in str:
4         if i in "aeiou":
5             return i
6
7 str = "hello worlde"
8 str = list(str)
9 print(str)
10 s = countofvowels(str)
11 str.remove(s)
12 s1 = str.index(s)
13 print(s1+1)
14
```

```
[ 'h', 'e', 'l', 'l', 'o', ' ', 'w', 'o', 'r', 'l', 'd', 'e']
11
```

question25**You work in the message encoding department of a national security**

agency. Every message

that is sent from or received in your office is encoded. you have a string containing , alpha numeric characters. of N is squared and the squares are concatenated together to encode the original number. Your task is to find and return an integer value representing the encoded value of the number. input1: An a string representing the number and chracters Output : Return an

In [16]:

```

1 def encode_message(input_string):
2     numeric_chars = [char for char in input_string if char.isdigit()]
3
4     squared_str = ''.join(str(int(char) ** 2) for char in numeric_chars)
5     encoded_value = int(squared_str)
6
7     return encoded_value
8
9 input_string = "hello 123 good morning"
10 encoded_value = encode_message(input_string)
11 print(f"Encoded value: {encoded_value}")

```

TypeError

Traceback (most recent call last)

```

Cell In[16], line 10
    7     return encoded_value
    9 input_string = "hello 123 good morning"
--> 10 encoded_value = encode_message(input_string)
    11 print(f"Encoded value: {encoded_value}")

Cell In[16], line 4, in encode_message(input_string)
    1 def encode_message(input_string):
    2     numeric_chars = [char for char in input_string if char.isdigit()]
--> 4     squared_str = ''.join(str(int(char) ** 2) for char in numeric_chars)
    5     encoded_value = int(squared_str)
    7     return encoded_value

Cell In[16], line 4, in <genexpr>(.0)
    1 def encode_message(input_string):
    2     numeric_chars = [char for char in input_string if char.isdigit()]
--> 4     squared_str = ''.join(str(int(char) ** 2) for char in numeric_chars)
    5     encoded_value = int(squared_str)
    7     return encoded_value

TypeError: 'list' object is not callable

```

question26

In [18]:

```
1 rows = int(input("Enter number of rows: "))
2
3 k = 0
4 count=0
5 count1=0
6
7 for i in range(1, rows+1):
8     for space in range(1, (rows-i)+1):
9         print(" ", end="")
10        count+=1
11
12    while k!=((2*i)-1):
13        if count<=rows-1:
14            print(i+k, end=" ")
15            count+=1
16        else:
17            count1+=1
18            print(i+k-(2*count1), end=" ")
19            k += 1
20
21    count1 = count = k = 0
22    print("")
```

Enter number of rows: 5

```
1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

In [22]:

```

1 rows = int(input("Enter number of rows: "))
2
3 total_sum = 0
4
5 for i in range(1, rows + 1):
6
7     for space in range(1, (rows - i) + 1):
8         print(" ", end="")
9
10
11    for j in range(i, 0, -1):
12        print(j, end=" ")
13        total_sum += j
14
15
16    for j in range(2, i + 1):
17        print(j, end=" ")
18        total_sum += j
19
20    print()
21
22 print("\nSum of the pyramid numbers:", total_sum)

```

Enter number of rows: 5

```

1
2 1 2
3 2 1 2 3
4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5

```

Sum of the pyramid numbers: 65

question27

Reduce till zero

Dev loves the number zero. Dev gives Andrew two integers X and Y and asks him to perform the steps below on X and Y. until the value of Y has been reduced to zero. The below steps should be followed sequentially:

1. If $X \leq 0$, then return X
2. If $Y=0$, then return X
3. Otherwise, let $T = X-Y$.
4. Set $X=Y$ and then set $Y=T$
5. Repeat from step 1. Your task is to help Andrew find and return an integer value, representing the value of X, when the value of Y has been reduced to zero. Note: At least one of the X or Y will be a non-zero integer Input Specification: input1: An integer value X, representing the first number. Input2: An integer value Y, representing the second number Sample input: 48 18 Sample Output: 6

question28

...

Equilibrium You are given an array A of N integers. An equilibrium position is a position where the sum of all integers on its left is equal to the sum of all integers on its right in the array A. Print the index of the equilibrium position. Note:For any given array there is only a single equilibrium position, if no equilibrium position is found then print "NOT FOUND" without quotes. The array is 1 indexed. Input Format: The input consists of two lines: The first line contains an integer denoting N. The second line contains N space-separated integers denoting the elements of the array A. Input will be read from the STDIN by the candidate Output Format: Print the index of the equilibrium position. If no index is found, print "NOT FOUND" Sample Input 5 2 4 3 2 7 Sample Output 3 "

In [5]:

```
1 def find_equilibrium_index():
2     import sys
3     input = sys.stdin.read
4     data = input().split()
5
6     N = int(data[0])
7     A = list(map(int, data[1:N+1])) # Ensure the list is correctly sliced
8
9     total_sum = sum(A)
10    left_sum = 0
11
12    for i in range(N):
13        total_sum -= A[i]
14        if left_sum == total_sum:
15            print(i + 1) # Return 1-indexed position
16            return
17        left_sum += A[i]
18
19    print("NOT FOUND")
20
21
22 # Example of how to test the function directly
23 if __name__ == "__main__":
24     import sys
25     from io import StringIO
26
27     # Replace this string with any input you want to test
28     test_input = "5\n2 4 3 2 7"
29     sys.stdin = StringIO(test_input)
30
31     find_equilibrium_index()
32
33
```

NOT FOUND

question29

In [1]:

```

1 def compute_gcd(x, y):
2     while(y):
3         x, y = y, x % y
4     return x
5
6 def compute_lcm(x, y):
7     gcd = compute_gcd(x, y)
8     return abs(x * y) // gcd
9
10 num1 = int(input("Enter the first number: "))
11 num2 = int(input("Enter the second number: "))
12 gcd = compute_gcd(num1, num2)
13 print("The G.C.D. is", gcd)
14 lcm = compute_lcm(num1, num2)
15 print("The L.C.M. is", lcm)

```

Enter the first number: 2
 Enter the second number: 4
 The G.C.D. is 2
 The L.C.M. is 4

question30

""

1. Signature for LCM Given two numbers a and b. Find the GCD and LCM of a and b. Input:

- Two positive integers a and b ($1 \leq a, b \leq 1000$) Output: For GCD function, an integer representing the GCD of a and b For LCM function, an integer representing the LCM of a and b Sample Input: 12 18 Output: 6 36

""

```
In [2]: 1 def find_missing_pangram_chars(input_string):
2     alphabet_set = set('abcdefghijklmnopqrstuvwxyz')
3
4     input_set = set(input_string.lower())
5
6     missing_chars = alphabet_set - input_set
7
8     missing_chars = sorted(missing_chars)
9
10    return ''.join(missing_chars)
11
12 input_string = "welcome to geeksforgeeks"
13
14 missing_characters = find_missing_pangram_chars(input_string)
15 print(missing_characters)
```

abdhijnpquvxyz

question31

```
In [4]: 1 def pangram(input_string):
2
3     alphabet = 'abcdefghijklmnopqrstuvwxyz'
4
5     input_string = input_string.lower()
6
7     missing_chars = []
8
9     for i in alphabet:
10         if i not in input_string:
11             missing_chars.append(i)
12
13     missing_chars.sort()
14     print(''.join(missing_chars))
15
16 input_string = input("Enter a string: ")
17 pangram(input_string)
```

Enter a string: priya
bcdefghjklmnoqstuvwxyz

question32

You are given a string containing words separated by spaces. Your task is to write a

function or program that reverses the order of words in the string. Sample Input: Hello World
Sample Output: World Hello

```
In [5]: 1 def rev(inp):
2     str=inp.split()
3     rev=str[::-1]
4
5     return ' '.join(rev)
6
7 inp_str=input("Enter a string: ")
8 rev_str=rev(inp_str)
9 print(rev_str)
```

Enter a string: hello world
world hello

```
In [7]: 1 str=input().split()
2 str=str[::-1]
3 print(*str,sep=" ")
```

hello world
world hello

question33

Peak Element Finder

Description: You are given an N- dimensional array arr[]. A peak element in the array is defined as an element whose value is greater than or equal to its neighboring elements (if they exist).

Your task is to find the index of any peak element in the given array Note: use 0-based indexing

Input: An integer representing the number of elements in the array. N space-separated integers, denoting the elements of the array. N space-separated integers ,denoting the elements of the array arr[] Sample Input: 5 1 3 20 4 1 Sample Output: 2

```
In [10]: 1 def peak(n):
2     n=len(arr)
3     if n == 1:
4         return 0
5     for i in range(1,n-1):
6         if arr[i-1]<=arr[i] and arr[i]>=arr[i+1]:
7             return i
8         if arr[i-1]>=arr[i-2]:
9             return i-1
10        else:
11            return i-2
12 N = int(input())
13 arr = list(map(int, input().split()))
14 index=peak(arr)
15 print(index)
```

4
1 20 3 4
1

question34

Number of toys

Akshay has a number of toys and he decided to donate some of them to an NGO. After the donation, he still has some toys left. Write a program to help Akshay to determine the number of remaining toys. Example: Input: 50 45 Output: The remaining number of toys = 5 Input: 60 6 Output: The remaining number of toys = 54

```
In [1]: 1 def remaining_toys(initial, donated):
2     return initial - donated
3
4 input_data = input().split()
5
6 initial_toys = int(input_data[0])
7 donated_toys = int(input_data[1])
8
9 remaining = remaining_toys(initial_toys, donated_toys)
10
11 print(f"The remaining number of toys = {remaining}")
```

```
50 45
The remaining number of toys = 5
```

question35

Smallest Number

Prince participated in three Olympiads at school and received marks for all of them. He is interested in finding out the lowest mark he obtained among the three Olympiads. Write a program to find the minimum mark. Example: Input: 50 66 23 Output: Smallest number is 23

```
In [4]: 1 def smallest(a,b,c):
2     if a<b and a<c:
3         return a
4     elif b<a and b<c:
5         return b
6     elif c<b and c<a:
7         return c
8 n1=int(input())
9 n2=int(input())
10 n3=int(input())
11 print(smallest(n1,n2,n3))
12
```

```
4
5
6
4
```

```
In [ ]: 1 input_data = input().split()
2 arr = list(map(int, input_data))
3 smallest = arr[0]
4
5 for i in range(len(arr)):
6     if arr[i] < smallest:
7         smallest = arr[i]
8
9 print(f"Smallest number is {smallest}")
10
```

question36

You are competing in a basketball contest. In this contest the score for each

successful shot depends on both the distance from the basket and the player's position. The ball is shot N times, successfully. You are given an array A containing the distance of a player from basket for N shots. The index of array represents the position of the player. Score is calculated by multiplying the position with the distance from the basket. Your task is to find and return an integer value, representing the maximum possible score you can achieve by choosing a contiguous subarray of size K from the given array. Note:

- A subarray is a contiguous part of array.
- Assume 1 based indexing.
- The array contains both negative and positive values.
- Assume the player is standing on a cartesian plane. Input Format
- input1:An integer value N representing the number of shots made by the player
- input2 : An integer K representing the size of subarray
- input3 : An array of integers Sample Input 5 2 1 2 3 4 5 Sample Output 14 S

```
In [1]: 1 def max_possible_score(N, K, A):
2     B = [(i + 1) * A[i] for i in range(N)]
3
4     current_sum = sum(B[:K])
5     max_sum = current_sum
6
7     for i in range(K, N):
8         current_sum += B[i] - B[i - K]
9         if current_sum > max_sum:
10             max_sum = current_sum
11
12     return max_sum
13 N = 5
14 K = 2
15 A = [1, 2, 3, 4, 5]
16
17
18 print(max_possible_score(N, K, A))
19
```

41

question37

sub array with max sum

You are given a list of integers, and your task is to find the subarray with the maximum sum.

Write a function or method to solve this problem efficiently and return the maximum sum. Input:

n: the no of elements in the array nums (List of integers): A list of integers ($1 \leq \text{len(nums)} \leq 10^5$) Sample input: 8 -1 2 3 10 -4 7 2 -5 Sample output: 20 Explanation: The max subarry sum is 20. The subarray is [2,3,10,-4,7,2]

In []:	1
---------	---

In []:	1
---------	---

In []:	1
---------	---