

Hacker earth Questions

1. Problem Statement: Little Robert likes mathematics. Today his teacher has given him two integers and asked to find out how many integers can divide both the numbers. Would you like to help him in completing his school assignment?

Input Formatting: There is two integers, a and b as input to the program.

Output Formatting: Print the number of common factors of a and b. Both the input value should be in a range of 1 to 10^{12} .

Sample:

Input: 10 15

Output: 2

Explanation: The common factors of 10 and 15 are 1 and 5. So the answer will be 2.

Solution:

```
data = input()
li = data.split()

a = int(li[0])
b = int(li[1])

def gcd(a, b):
    if (a == 0):
        return b;
    return gcd(b%a, a);

if (a>0 and a<(10**12+1) and b>=1 and b<(10**12+1)):
    count = 1
    for i in range(2, gcd(a, b)+1):
        if a%i==0 and b%i==0:
            count = count+1
    print(count)
```

Explanation:

We are reading two integers as a single input and then splitting it using the `split()` method.
The function `gcd()` is to find the greatest common divisor.

2. Problem Statement: Consider a permutation of numbers from 1 to N written on a paper. Let's denote the product of its element as 'prod' and the sum of its elements as 'sum'. Given a positive integer N, your task is to determine whether 'prod' is divisible by 'sum' or not.

Input Format: First input will be an integer T. It depicts a number of test cases. Followed by value for each test case. Each test case will contain an integer N ($1 \leq N \leq 10^9$). It is nothing but the length of the permutation.

Output Format: For each test case, print "YEAH" if 'prod' is divisible by 'sum', otherwise print "NAH".

Solution:

```
testSize = int(input())
nArr=[]
for i in range(1,testSize+1):
    nArr.append(int(input()))

for n in nArr:
    if n>=1 and n<=(10**9):
        prod = 1
        sum = 0
        for i in range(1, n+1):
            prod = prod*i
            sum = sum+i
        if prod%sum==0:
            print("YEAH")
        else:
            print("NAH")
```

Sample:

Input:

2
2
3

Output:

YEAH

NAH

Explanation:

The first input is a number of test cases.

Iterate to read all the inputs and store it in the Python list.

For each element in the list, calculate prod and sum.

Print “YEAH” if the prod is divided by the sum. Otherwise, print “NAH”.

3. Problem Statement:

Group Anagrams- You have given a list of words. Write a program to returns a list of sets of anagrams.

Example:

Input: ['cat', 'dog', 'god']

Output: [{'cat'}, {'dog', 'god'}]

Input= ['cat', 'dog', 'cat', 'god']

Output= [{'cat'}, {'dog', 'god'}]

Solution:

```
def getAnagramSet(words):
```

```
    #remove duplicate words from the list
```

```
    words=list(set(words))
```

```
    #find the anagram of each word in the list
```

```
anagramWords=[''.join(sorted(x)) for x in words]

finalOut=[]
for word in set(anagramWords):
    #find the index for all occurrences of string word
    ind=[i for i, x in enumerate(anagramWords) if x==word]

    #make the list of all the string
    #which is anagram to each other
    tempOut=[]
    for i in ind:
        tempOut.append(words[i])

    #append all the set to the final output list
    finalOut.append(set(tempOut))

return finalOut

words=['cat', 'dog', 'god', 'cat']

if __name__ == "__main__":
    print(getAnagramSet(words))
```

4. Problem Statement:

An array contains n numbers ranging from 0 to $n-1$. There are some numbers duplicated in the array.

It is not clear how many numbers are duplicated or how many times a number gets duplicated.

How do you find a duplicated number in the array?

Example:

If an array of length 7 contains the numbers {2, 3, 1, 0, 2, 5, 3}, the implemented function (or method) should return either 2 or 3.

Solution:

```
def findDup(liArr):  
    liArr.sort()  
  
    liDuplicate=[]  
    for i in range(0, len(liArr)-1):  
        if liArr[i]==liArr[i+1]:  
            liDuplicate.append(liArr[i])  
  
    return liDuplicate  
  
print(findDup([2, 3, 1, 0, 2, 5,3]))
```

5. Problem statement:

You have given an array (list in case of Python). Find the index of the element in the array such that the sum of all the elements left to the index is equal to the sum of all the elements right to the index. (Split the array into equal sum subarrays)

Example:

Input (given array list):
[2, 4, 5, 1, 2, 3]

Output:
2

Explanation:

The left and right sub array for the index "2" is [2, 4] and [1, 2, 3]. The sum of both the sub array is 6.

Solution:

```
sample = [2, 4, 5, 1, 2, 3]

sum = 0

sum_left=[]
for val in sample:
    sum += val
    sum_left.append(sum)

sum_right=[]
for val in sample:
    sum_right.append(sum)
    sum -= val

for i in range(len(sum_left)):
    if sum_left[i] == sum_right[i]:
        print(f"Matching index is {i}.")
        break
```