

---

## Ant on rail

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 Sample Input

5

1 -1 1 -1 1

**Sample Output:**

2

```
steps=int(input())#5
```

**Solution:**

```
a=list(map(int,input().split()))
```

```
sp=0
```

```
ans=0
```

```
for i in a:
```

```
    sp=sp+i
```

```
    if sp==0:
```

```
        ans=ans+1
```

```
print(ans)
```

Output

5

1 -1 1 -1 1

2

---

## chocolate jar

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 find 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.

### Solution:

```
jars=list(map(int,input().split()))
n=int(input())
a=0
for i in jars:
    print("this iteration is for",i)
    a=a+(i//3)
    print(a)
    #remainder
    if i%3!=0:
        a=a+1
    else:
        a=a+0
print(a)
```

### Output

```
10 20 30
3
this iteration is for 10
3
this iteration is for 20
10
this iteration is for 30
21
21
```

## Dog Age

### Input format:

Input1 :an integer value n representing the age of max's dog

### Output format:

Return an integer value representing the age of

### **Solution:**

```
n=int(input())
age=n*7
print(age)
```

### **Output:**

```
50
350
```

## Diwali party

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

Your task is help Max find and return an integer value, representing the number of problems Max can solve and reach the party venue within the given time frame of 4 hours.

Note: Max will leave his home at exactly B PM to reach the party venue.

### Input Format:

Input: An integer value N, representing the total number of problems.

Input2: An integer value P, Representing the time to travel in minutes from his home to the party venue.

### Solution:

```
probs=int(input())
tot=int(input())
c=0
s=0
rt=4*60-tot
for i in range(1,probs+1):
    s=s+5*i
    if s>rt:
        break
    c=c+1
for i in range(1,probs+1):
    if s<rt:
        s=s+5*i
        c=c+1
print(c)
```

### Output:

```
6
180
4
```

## Basketball

You are competing in a basketball .in this contest the score for each successful shot depends on both the distance from the basket and 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 & 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

Input: An Integer value representing the number of shots made by the player

Input: As Integer & representing the size of subarray

Input: An array of integers

### Solution:

```
input1=int(input())
input2=int(input())
arr=list(map(int,input().split()))
mx=-1
for i in range(0,len(arr)-input2+1):
    temp=arr[i:i+input2]
    k,s=1,0
    for j in temp:
        s+=(j*k)
        k+=1
    if s>mx:
        mx=s
print(mx)
```

### Output:

```
3
5
2 -1 3 5 -2
19
```

## Space counter

You have been given the task of making the counter on a social media platform more user-friendly. Your task is to find and return an integer value representing the count of the number of spaces in a given string S.

### Input:

A string:

### Output:

Return an integer value representing the count of the number of space in a given string S.

### Solution:

```
#without using split
s=input()
c=0
for i in s:
    if i==" ":
        c=c+1
# with split
h=s.split()
print(h)
print(len(h)-1)
```

### Output :

```
hi hello hey
['hi', 'hello', 'hey']
2
```

## Encoded the number

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 an integer N, and each digit 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.

### Input format

Input1: An integer value N representing the number to be encoded

### Output :

Return an integer value representing the encoded value of the number

### Solution:

```
n=167
#sod
def sod(n):
    c=0
    while n>0:
        c=c+1
        n=n//10
    return c

def rev(n):
    ans=0
    while n>0:
        digit=n%10
        sq=digit**2
        sod_sq=sod(sq)
        ans=ans*(10**sod_sq)+sq
        n=n//10
    return ans
ans=rev(n)

def rev2(n):
    ans2=0
    while n>0:
        digit=n%10
        ans2=ans2*10+digit
        n=n//10
    return ans2
print(rev2(ans))
```

### Output:

16394

## Farthest 1

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.

### Solution:

```
a=list(map(int,input().split()))
c=-1
s=0
for i in a:
    s=s+i
    if s>c:
        c=s
print(c)
```

### Output:

```
1 2 6 -5
9
```



## Minimum array sum

Paul is given an array A of length N. He must perform the following operation on the array sequentially:

- Choose any two integers from the array and calculate their average
- If an element is less than the average, update it to 0. However, if element is greater than or equal to average, he need not update it.
- Your task is to help Paul find and return an integer value, representing the minimum 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

### Input format:

Input1: An integer value N, representing the size of the array A.

Input2 :An integer array A.

### Output format:

Return an integer value representing The minimum possible sum of all the elements in the array by

### Solution:

```
a=list(map(int,input().split()))
mx=-999
mx2=-999
```

```
#maxi
for i in a:
    if i>mx:
        mx=i
#maxi2
for i in a:
    if i>mx2 and i!=mx:
        mx2=i
avg=(mx+mx2)/2
ans=0
for i in a:
    if i>avg:
        ans+=i
print(ans)
print(mx2)
```

### Output

```
12 16 22 2
22
16
```

## 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 time.

### Sample input:

HHHTT

### Solution:

```
s=input()
hc=0
tc=0
score=0
for i in s:
    if i=="H" or i=="h":
        tc=0
        hc+=1
        score+=2
        if hc==3:
            break
    else:
        hc=0
        tc+=1
        score-=1
        if tc==3:
            break
print(score)
```

### Output:

HHHTT

6

## Find missing and repeated value

You are given a 0-indexed 2D integer matrix grid of size  $n * n$  with values in the range  $[1, n^2]$ . Each integer appears exactly once except a which appears twice and b which is missing. The task is to find the repeating and missing numbers a and b. Return a 0-indexed integer array ans of size 2 where ans[0] equals to a and ans[1] equals to b.

### Solution:

```
a=[]
r=int(input())
for i in range(0,3):
    sub=[]
    print("enter the value for row",i)
    for j in range(0,3):
        print("enter the value for column")
        ele=int(input())
        sub.append(ele)
    print(sub)
    a.append(sub)
print(a)
```

```
d={}
ans=[]
for i in range (0,r):
    for j in range(0,c):
        if a[i][j] not in d:
            d[a[i][j]]=1
        else:
            d[a[i][j]]+=1
            if d[a[i][j]]==2:
                ans.append(a[i][j])
print(d)
for i in range(1,r**2+1):
    if i not in d:
        ans.append(i)
print(d)
print(ans)
```

### Output:

```
3
enter the value for row 0
enter the value for column
1
```

```
[1]
enter the value for column
2
[1, 2]
enter the value for column
3
[1, 2, 3]
[[1, 2, 3]]
enter the value for row 1
enter the value for column
```

## Find the sum of encrypted

You are given an integer array `nums` containing positive integers. We define a function `encrypt` such that `encrypt(x)` replaces every digit in `x` with the largest digit in `x`. For example, `encrypt(523) = 555` and `encrypt(213) = 333`. Return the sum of encrypted elements.

### Example 1:

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

output=6

Explantation:the encrypted element are[1,2,3].the sum of encrypted elements is  $1+2+3==6$

### Solution:

```
n=list(map(int,input().split()))
def encrypt(n):
    mx=0
    c=0
    ans=0
    while temp>0:
        d=temp%10
        if d>mx:
            mx=d
        temp//=10
        c=c+1
        n=n//10
    while c>0:
        ans=ans*10+mx
        c=c-1
    return(ans)
```

### Output:

## 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 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

### Solution:

```
a=input()
s=input()
total=0
for i in a:
    if i not in s:
        temp=125
        for j in s:
            d=abs(ord(i)-ord(j))
            if d<temp:
                temp=d
        total+=temp
print(total)
```

### Output:

abcd

xyz

86

## Pyramid sum

Adam has a pyramid of numbers. The pyramid structure is formed by arranging the numbers in the following pattern

```
1
212
32123
4321234
543212345
```

N\_\_212\_\_N

- The first row contains the number 1. The second row contains the numbers 2, 1, and 2. The third row contains the numbers 3, 2, 1, 2, and 3. This pattern continues for subsequent rows, until it reaches N, which represents the height of the pyramid.
- Given height of pyramid N find the sum of the numbers in the pyramid and return the sum as the output.

### Sample input:

3

### Sample output:

17

### Explanation:

```
1
212
32123
```

sum=1+2+1+2+3+2+1+2+3=17

### Solution:

```
rows=int(input()) #4
ans=0
for i in range(2,rows+1): #2,3,4
    n=i #2
    while n>1: #2 #3,2 #4
        ans +=2*n #2*2=4 #4+2*3 #4+2*3+2*2=14 #14+2*4 #14+2*4
        n=n-1 #1 #2 #3
    ans+=rows #3
```

### Output:

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

## Vowel repetition problem

Given a string s print the most frequent vowel that is present in the string as a output

### Input Format:

A single line containing the string s

The input will be read form the STDIN by the candidate

### Output:

Print a single character which represent the most frequent vowel in the given string

### Solution:

```
s=input()
v="aeiou"
mx=-999
ans=0
d={}
for i in s:
    if i in v:
        if i not in d:
            d[i]=1
        else:
            d[i]+=1
        if d[i]>mx:
            mx=d[i]
            ans=i
print(d)
print(ans)
```

### Output:

```
helloworld
{'e': 1, 'o': 2}
o
```



## Prime factors

prime factors of a positive integer are the prime numbers that divide that integer exactly.

given an array arr of n integer and a positive integer num.

let's suppose prime factorization of num is

Sum of numbers in array arr at indices of prime factors of number num is:

### Output format:

Print the sum that was mention in the problem statement

### Solution:

```
def pf(n):
    ans=[]
    i=2
    while i<=n:
        if n%i==0:
            ans.append(i)
            n=n//i
        else:
            i=i+1
    return ans
ans=pf(6)
s=0
a=[11,12,13,14,15,16]
for i in ans:
    s=s+a[i]
print(s)
```

### Output:

13

## 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

### Solution:

```
a,y=map(int,input().split())
ans=0
while True:
    if a%y==0:
        ans=y
        break
    else:
        y=y+1
s=0
while ans>0:
    digit=ans%10
    s=s+digit
    ans=ans//10
print(s)
```

### Output:

100 17

2

## Equilibrium position

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"

### Solution:

```
n=input()
a=list(map(int,input().split()))
f=0
for i in range(0,len(a)):
    s1=0
    s2=0
    i1=i
    j=i+1
    while i>=0:
        s1+=a[i]
        i=i-i1
    while j<len(a):
        s2+=a[j]
        j=j+1
    if( s1==s2):
        print(i1+1)
        f=1
        break
if f==0:
    print(len(a)//2)
```

## Space-separated

**Note:**use 0-based indexing

**Input:**

An integer repressing the number of element in the array.N space-separated integers, denoting the element of the array

N space-separated integers, denoting the elements of the array arr[]

**Solution:**

```
a=list(map(int,input().split()))
mp=-999
for i in range(1,len(a)-1):
    if a[i]>a[i-1]and a[i]>a[i+1]:
        p=a[i]
        if p>mp:
            mp=p
print(mp)
```

**Output:**

1 3 20 4 1

20

## Triplets

### Input:

7

5 3 20 10 1 4 2

60

### Output:

3

### Explanation:

Product m:60

Possible triplets for product m:(5,4,3),(20,3,1),(10,3,2)

The count of unique triplets is 3

### Solution:

```
t=60
```

```
c=0
```

```
pro=1
```

```
a=[5,3,20,10,1,4,2]
```

```
for i in range(0,len(a)):
```

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

```
        for k in range(i+1,len(a)):
```

```
            print("indexes",i,j,k)
```

```
            print("value",a[i],a[j],a[k])
```

```
            pro=a[i]*a[j]*a[k]
```

```
            if pro==t:
```

```
                print(pro)
```

```
                print("triplet",a[i],a[j],a[k])
```

```
                c=c+1
```

```
        print()
```

```
print(c)
```

### Output

```
indexes 0 1 1
```

```
value 5 3 3
```

```
indexes 0 1 2
```

```
value 5 3 20
```

```
indexes 0 1 3
```

```
value 5 3 10
```

```
indexes 0 1 4
```

```
value 5 3 1
```

indexes 0 1 5  
value 5 3 4  
60  
triplet 5 3 4

indexes 0 1 6  
value 5 3 2

indexes 0 2 1  
value 5 20 3

indexes 0 2 2  
value 5 20 20

indexes 0 2 3  
value 5 20 10

indexes 0 2 4  
value 5 20 1

indexes 0 2 5  
value 5 20 4

indexes 0 2 6  
value 5 20 2

indexes 0 3 1  
value 5 10 3

indexes 0 3 2  
value 5 10 20

indexes 0 3 3  
value 5 10 10

indexes 0 3 4  
value 5 10 1

indexes 0 3 5  
value 5 10 4

indexes 0 3 6  
value 5 10 2

indexes 0 4 1

value 5 1 3

indexes 0 4 2

value 5 1 20

indexes 0 4 3

value 5 1 10

indexes 0 4 4

value 5 1 1

indexes 0 4 5

value 5 1 4

indexes 0 4 6

value 5 1 2

indexes 0 5 1

value 5 4 3

60

triplet 5 4 3

indexes 0 5 2

value 5 4 20

indexes 0 5 3

value 5 4 10

indexes 0 5 4

value 5 4 1

indexes 0 5 5

value 5 4 4

indexes 0 5 6

value 5 4 2

indexes 0 6 1

value 5 2 3

indexes 0 6 2

value 5 2 20

indexes 0 6 3

value 5 2 10

indexes 0 6 4  
value 5 2 1

indexes 0 6 5  
value 5 2 4

indexes 0 6 6  
value 5 2 2

indexes 1 2 2  
value 3 20 20

indexes 1 2 3  
value 3 20 10

indexes 1 2 4  
value 3 20 1  
60  
triplet 3 20 1

indexes 1 2 5  
value 3 20 4

indexes 1 2 6  
value 3 20 2

indexes 1 3 2  
value 3 10 20

indexes 1 3 3  
value 3 10 10

indexes 1 3 4  
value 3 10 1

indexes 1 3 5  
value 3 10 4

indexes 1 3 6  
value 3 10 2  
60  
triplet 3 10 2

indexes 1 4 2  
value 3 1 20  
60



triplet 3 1 20

indexes 1 4 3  
value 3 1 10

indexes 1 4 4  
value 3 1 1

indexes 1 4 5  
value 3 1 4

indexes 1 4 6  
value 3 1 2

indexes 1 5 2  
value 3 4 20

indexes 1 5 3  
value 3 4 10

indexes 1 5 4  
value 3 4 1

indexes 1 5 5  
value 3 4 4

indexes 1 5 6  
value 3 4 2

indexes 1 6 2  
value 3 2 20

indexes 1 6 3  
value 3 2 10  
60  
triplet 3 2 10

indexes 1 6 4  
value 3 2 1

indexes 1 6 5  
value 3 2 4

indexes 1 6 6  
value 3 2 2

indexes 2 3 3  
value 20 10 10

indexes 2 3 4  
value 20 10 1

indexes 2 3 5  
value 20 10 4

indexes 2 3 6  
value 20 10 2

indexes 2 4 3  
value 20 1 10

indexes 2 4 4  
value 20 1 1

indexes 2 4 5  
value 20 1 4

indexes 2 4 6  
value 20 1 2

indexes 2 5 3  
value 20 4 10

indexes 2 5 4  
value 20 4 1

indexes 2 5 5  
value 20 4 4

indexes 2 5 6  
value 20 4 2

indexes 2 6 3  
value 20 2 10

indexes 2 6 4  
value 20 2 1

indexes 2 6 5  
value 20 2 4

indexes 2 6 6

value 20 2 2

indexes 3 4 4

value 10 1 1

indexes 3 4 5

value 10 1 4

indexes 3 4 6

value 10 1 2

indexes 3 5 4

value 10 4 1

indexes 3 5 5

value 10 4 4

indexes 3 5 6

value 10 4 2

indexes 3 6 4

value 10 2 1

indexes 3 6 5

value 10 2 4

indexes 3 6 6

value 10 2 2

indexes 4 5 5

value 1 4 4

indexes 4 5 6

value 1 4 2

indexes 4 6 5

value 1 2 4

indexes 4 6 6

value 1 2 2

indexes 5 6 6

value 4 2 2

6

## 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 ( $1 \leq \text{len}(\text{nums}) \leq 10^5$ )

### Sample Input:

8

11,12,13,145

-12310-472-5

2230

### Sample output:

5115

20

27

### Explanation:

The max subarry sum is 20. The subarray is (2,3,10,-4,7.2)

### Solution:

```
a=list(map(int,input().split()))
c_sum=0
mx_sum=0
for i in a:
    c_sum+=i
    if c_sum<0:
        c_sum=0
    if c_sum>mx_sum:
        mx_sum=c_sum
print(mx_sum)
```

### Output

2 3 10 -4 7 2

20

## Target sum

You are given a list of integers, and your task is to write a function that finds the two numbers in the list that add up to a specific target sum. You need to return the indices of these two numbers.

Write a function that takes a list of Integers and a target sum as input and returns a list of two indices (0-based) of the numbers that add up to the target sum. Assume that there is exactly one solution, and you cannot use the same element twice

### Sample Input:

```
11 15
9
27
```

### Sample Output:

```
[0,1]
```

### Solution:

```
a=list(map(int,input().split()))
t=int(input())
a.sort()
i=0
j=len(a)-1
ans=0
while i<j:
    curr_sum=a[i]+a[j]
    if curr_sum==t:
        print(i,j)
        i+=1
        j-=1
    elif curr_sum<t:
        i=i+1
    else:
        j=j-1
```

### Output

```
2 7 11 15
9

0 1
```

## 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

### Solution:

```
s=input()
i=0
c=0
while i<len(s)-1:
    if s[i]=="0" and s[i+1]=="0":
        c +=1
        i +=2
    else:
        c +=1
        i +=1
if i<len(s):
    c +1
print(c)
```

### Output 1:

100

2

### Output 2:

10010010

5

## Magic string

Eve has a string  $S$  containing lowercase English letters. She wants to transform this string into a Magic String, where all the characters in the string are the same. To do so, she can replace any letter in the string with another letter present in that string.

Your task is to help Eva find and return an integer value, representing the minimum number of steps required to form a Magic String. Return 0, if  $S$  is already a Magic String

### Input Specification:

Input: A string  $S$ , containing lowercase English letters.

### Output Specification:

Return an integer value, representing the minimum number of steps required to form a Magic String. Return 0, If  $S$  is already a Magic String

### Sample input:

aaabbbccddddd

### Sample output:

8

### Solution:

```
s=input()
d={}
mx=-999
for i in s:
    if i in d:
        d[i]+=1
    else:
        d[i]=1
    if d[i]>mx:
        mx=d[i]
print(len(s)-mx)
```

### Output1:

aaabbbccddddd

8

### Output2:

aaaahhhh

4

## pattern

Print the given pattern

You are given a number n and you have to print the given pattern:

For n=3

3 3 3 2 2 2 1 1 1

3 3 2 2 1 1

3 2 1

### Solution:

N = 3

k=0

for i in range(N,0,-1):

    for j in range(N, 0, -1):

        k=i

        while k>0:

            print(j,end=" ")

            k=k-1

    print()

### Output:

3 3 3 2 2 2 1 1 1

3 3 2 2 1 1

3 2 1



## Missing Alphabets

Pangram is a sentence containing every letter in the English alphabet. Given a string, find all characters that are missing from the string, Le., the characters that can make the string a Pangram. We need to print output in alphabetic order.

**For example,**

Input: welcome to geeksforgeeks

Output: abdhijnpquvxyz

**Solution:**

```
org="qwertyuiopasdfghjklzxcvbnm"
s=input()
ans=""
for i in org:
    if i not in s:
        ans=ans+i
print(ans)
```

**Output:**

hello

qwrtyuipasdfgjkzxcvbnm

## Dev loves the number 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  $X \leq Y$  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:

Input 1: An Integer value X. representing the first number.

Input2: An integer value Y, representing the second number

### Sample Input:

48

18

### Solution:

```
x=int(input())
y=int(input())
while y>0:
    if x<y:
        temp=x
        x=y
        y=temp
    #x,y=y,x
    t=x-y
    x=y
    y=t
print(x)
```

### Output:

48

18

6

## Fellis Function

Morris Fellis has come up with a new function called Fellis function Morris defines the function as follows:

$$f(0) = 1$$

$$f(1) = 1$$

$$f(N) = f(N-1) + 7^N * f(N-2) + (N / 4) \text{ modulo } 10^9 + 7$$

Given an integer N, your task is to help Morris find and return an Integer value of f(N), after performing Fellis Function.

**Note:** Here the division operator is integer division operator ie, it divides two numbers and returns the integer part of the result

### Input Specification:

Input1: An integer. Value N representing the fellis function value.

### Solution:

```
n=int(input())
x=[1,1]
for i in range(2,n+1):
    ans=(x[i-1]+7*x[i-2]+i//4)%(10**9+7)
    x.append(ans)
print(x[n])
```

### Output:

```
8
6713
```

### Recursion method

```
def fel(n):
    if n==0:
        return 1
    if n==1:
        return 1
    return(fel(n-1)+7*fel(n-2)+n//4)%(10**9+7)
n=int(input())
print(fel(n))
```

### Output:

```
8
6731
```

## 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. Corner 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 separated strings showing the seat positions and gaps

Sample input:

3C

1A 2B-3C 4D

### Solution:

```
chair=input()
s=list(input().split())
z=999
c_ind=s.index(chair)
#TILL C_IND
for i in range(0,c_ind):
    if s[i]=="-":
        if abs(c_ind-i)-1<z:
            z=abs(c_ind-i)-1
        #RIGHT SIDE
for i in range (c_ind+1,len(s)):
    if( s[i]=="-"):
        if abs(i-c_ind)-1<z:
            z=abs(i-c_ind)-1
print(z)
```

### Output:

3C

1A 2B - 3C - 4D

0

## Special Fibonacci

Alex is exploring a series and she came across a special series, in which

$$f(N)=f(N-1)*f(N-1)+(N-2)*(N-2) \bmod 47$$

where  $f(0) = 1$ .  $f(1)=1$

Your task is to help Alex find and return an integer value, representing the Nth number in this special series.

### Input Specification:

input1: An integer value N.

### Output Specification:

Return an integer value, representing the Nt number in this special

### Solution:

```
def sfib(n):
    if n==0:
        return 1
    if n==1:
        return 1
    return(sfib(n-1)**2+(n-2)**2)%47
print(sfib(6))
```

Output:

34

Second method

```
x=[1,1]
n=int(input())
for i in range(2,n+1):
    ans=(x[i-1]**2+(i-2)**2)%47
    x.append(ans)
print(x[n])
```

Output:

6  
34

## Maximize Pair Product

Noah is given an integer array A of length N. He must perform the following operations:

- Select any Integer pair having sum equal to 18 from the array.
- Select the pair with maximum product such that the first element of the pair is greater than the second element of the pair.
- Your task is to help Noah find and return a pair in the form of an integer array, which satisfies the mentioned conditions.

### Input Specification:

Input1: An Integer value N, representing the size of array A.

Input2: An integer array A.

### Output Specification:

Return a pair in the form of an integer array, which satisfies the mentioned

### Sample input:

8

### Sample output:

### Solution:

```
a=list(map(int,input().split()))
p_max=-999
k=0
h=0
for i in range(0,len(a)):
    for j in range(i+1,len(a)):
        if a[i]+a[j]==18:
            if a[i]>a[j]:
                p=a[i]*a[j]
                if p>p_max:
                    p_max=p
                    k=a[i]
                    h=a[j]
print(p_max,k,h)
```

### Output:

11 12 2 8 10 11 9 8

80 10 8

**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
```

**Solution:**

```
a=[1,2,3]
b=[3,4,5]
mini=999
for i in range(len(a)):
    if a[i]+b[i]<mini:
        mini=a[i]+b[i]
print(mini)
```

**Output:**

```
4
```

## Prefix and suffix

You are given an array A of N integers. The array A can be divided into two parts: the first part consists of the first 'i' elements of A (where i ranges from 1 to N), and the second part consists of the last (N-i) elements of A

- Your task is to find and return a new array named result of the same size as A, where each element of result[i] represents the absolute difference between the sum of the elements in the first part of A and the sum of the elements in the second part of A

**Note:** For  $i = N, N-i = 0$ . So, consider the sum of last N-i integers as 0 in this case

### Input Specifications:

input1: An integer value representing the size of the array A.

Input2: An integer array A.

### Output Specification:

Return a new integer array named result of the same size as A, where each element of result(i) represents the absolute difference between the sum of the elements in the first part A and the sum of the elements in the second part of A

### Sample Input:

5

12345

### Sample Output:

[13, 9, 3, 5, 15]

### Solution:

```
a=list(map(int,input().split()))
ts=0
ls=0
rs=0
cs=0
ans=[]
for i in a:
    ts+=1
#left righth diff
for i in a:
    ls+=i
    rs=ts-ls
    cs=abs(rs-ls)
    ans.append(cs)
print(ans)
```

Output:



## Boring array

You are given an array 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:

Input. An integer value  $N$ , representing the size of array  $A$

Input? An integer array  $A$

### Output Specification:

Return an integer value, representing the maximum score

### Sample Input:

```
4
1 2 3 4
```

### Solution:

```
a=list(map(int,input().split()))
a.sort()
```

```
left=0
right=len(a)-1
s=0
while left<=right:
    d=abs(a[right]-a[left])
    s+=d
    left+=1
    right-=1
print(s)
```

Output:

```
1 2 3 4

4
```

## Solve the equation

Given an integer N, your task is to find and return the number of sets of 3 positive integers a, b and c. that satisfy the following equation:

$$a^2+b^2+c^2+ab+bc+ca=N$$

**Note:** a, b and c are positive integers, and their values can be the same.

### Input Specification:

input1: An integer value N

### Output Specification:

Return an integer value, representing the number of sets of three positive integers that satisfy the equation given above.

### Sample Input:

6

### Sample Output:

1

### Solution:

```
n=6
f=0
for a in range(1,n):
    for b in range(1,n):
        for c in range(1,n):
            if(a**2+b**2+c**2+a*b+b*c+c*a)==n:
                print(a,b,c)
                f=1
if f==0:
    print("NP")
```

### Output:

1 1 1

## Finding commas

Liam works as a data analyst for a company that stores massive amounts of numerical data. He has been tasked with determining how many commas are used when writing numbers in the range of 1 to N (inclusive) in a specific format

In this format, if numbers are more than four digits long, commas are used to separate the numbers into groups of three, starting from the right for the representation of the number. Your task is to help Liam find and return an integer value, representing the total number of commas used when writing each integer in the range of 1 to N

### Input Specification:

Input: An integer value N. representing the number range.

### Output Specification:

Return an integer value, representing total number of commas used when writing each integer in the range of 1 to N.

### Sample Input:

5000

### Sample Output:

4001

### Solution:

```
n=int(input())
c=1000
comma=1
res=0
while c<=n:
    m=c*1000
    num=min(n-c+1,m-c)
    res+=num*comma
    c=m
    comma+=1
print(res)
```

### Output:

1010000

1019002

## Robo Race

There is a robot race happening between two robots named Robotop and Robocop. Both the robots reach the starting point to begin the race on a Circular track

- Race starts at time  $T = 0$  seconds. Robotop starts the race at  $T = X$ th second and takes exactly  $N$  seconds to complete one lap. On the other hand, Robocop starts the race at  $T = Y$ th second and takes exactly  $M$  seconds to complete one lap.
- Your task is to find and return an integer value, representing the least time  $T$  (in seconds) at which these two robots meet each other again at the starting point.

### Sample Input:

2314

### Sample Output:

5

### Explanation:

$x = 2$   $N = 3$   $Y = 1$   $M = 4$

Robotop starts at  $T = 2$  and completes one lap every 3 seconds. Robocop starts at  $T = 1$  and completes one lap every 4 seconds. The smallest point where both meet at the starting point is 5 seconds.

### Solution:

```
x,m,y,n=map(int,input().split())
time=max(x,y)
while True:
    if time>=x and (time-x)%m==0 and time>=y and (time-y)%n==0:
        print(time)
        break
    time+=1
```

Output:

2 3 1 4

5

## Generated numbers

You have a jar which initially contains N marbles. You can perform the below operations in any order:

1. Taking out A number of marbles from the jar.
2. Taking out B number of marbles from the jar.

Your task is to find and return an integer value, representing the total number of unique positive number of marbles that can be left behind by performing these operations, including the initial number of marbles.

### Note:

You can perform the above operations any number of times and in any order keeping in mind that the jar should never become empty.

### Input Specification:

A single line containing space separated integers N,A,B

### Solution:

```
n=10
a=3
b=5
a1=n//a
b1=n//b
c=0
for i in range(a1+1):
    for j in range(b1+1):
        if i*a+j*b<10:
            c+=1
print(c)
```

### Output:

6

## Reverse the order of string:

You are given a string containing words separated by space. your task is to write a function or program that reverses the order of words in the string

### Solution 1:

```
s=input().split()
s=s[::-1]
print(*s,sep=" ")
```

### Solution:

```
i=0
j=len(l)-1

while i<=j:
    temp=l[i]
    l[i]=l[j]
    l[j]=temp
    i+=1
    j-=1
print(' '.join(l))
```

### Output:

hello

hello

olleh

## Best grade

Andrew has a string consisting of lowercase English letters representing respective grades of to students in his class. His grade is at Pth index. He can swap any two adjacent grades

- Your task is to help Andrew find and return a string value, representing maximized grade by bringing lexicographically smallest character on the Pth index after doing at most K swaps

### Sample Input:

abcdefg

3

2

### Sample Output:

a

### Solution:

```
s=input()
p=int(input())
k=int(input())
s1=(list(s))
s=0
e=len(s1)
mini=999
if abs(p-k-1)>=0:
    s=abs(p-k-1)
if p+k<len(s1):
    e=p+k
print(s,e)
for i in range(s,e):
    mini=min(ord(s1[i]),mini)
store=s1[p-1]
s1[p-1]=s1[s1.index(chr(mini))]
s1[s1.index(chr(mini))]=store
print(mini)
print(''.join(s1))
```

### Output:

abcdef

3

2

0 5

97

cbadef

### 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

#### Solution:

```
a=int(input())
b=int(input())
def gcd(a,b):
    while b!=0:
        temp=a
        a=b
        b=temp%b
    return a
def lcm(a, b):
    return (a*b)//gcd(a,b)
g=gcd(a,b)
l=lcm(a,b)
print(g,l)
```

#### Output:

12

18

6 36



### **Smallest number:**

Prince participated in three olympiads at school and received marks for all of them he is interested In finding out the lowest marks he obtained among the three olympiads.write a program to find the minimum marks

#### **Example:**

Input: 50 66 23

Output: smallest number is 23

#### **Solution:**

```
a,b,c=list(map(int,input().split()))
small=0
for i in (0,a):
    if a<b and a<c:
        small=a
    if b<c and b<a:
        small=b
    if c<a and c<b:
        small=c
print(small)
```

#### **Output:**

50 66 23

23

