1. Flipping bits

#!/bin/python3

import math

import os

import random

import re

import sys

def flippingBits(n):

    # Write your code here

    s=bin(n)

    a=s.replace("0b","")

    p=32-len(a)

    a=list(a)

    q=""

    for i in range(0,p):

        a.insert(0,'0')

    for i in range(0,32):

        if a[i]=='0':

            q+='1'

        else:

            q+='0'

    return int(q,2)

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    q = int(input().strip())

    for q\_itr in range(q):

        n = int(input().strip())

        result = flippingBits(n)

        fptr.write(str(result) + '\n')

    fptr.close()

# Valid PAN format

numbers="0123456789"

upper\_case="ABCDEFGHIJKLMNOPQRSTUVWXYZ"

n=int(input())

s=[]

for h in range(0,n):

s.append(input())

for j in range(0,n):

c=0

k=s[j]

l=len(k)

if l==10:

for i in range(0,l):

if i<=4 and k[i] in upper\_case:

c+=1

if i>4 and i<10 and k[i] in numbers:

c+=1

if i==10 and k[i] in upper\_case:

c+=1

if c==9:

print("YES")

else:

print("NO")

1. Encryption

#!/bin/python3

import math

import os

import random

import re

import sys

def encryption(s):

s.replace(" ","")

q = len(s)

w = q\*\*0.5

d,m=math.modf(w)

if d==0.0:

w=int(w)

else:

w+=1

w=int(w)

a=[]

j=0

new=""

for i in range(0,w):

f=j+w

a.append(s[j:f])

print(a,j,f)

j+=w

if ""==a:

a.remove("")

print(a)

for j in range(0,w):

for t in range(0,w):

try:

k=a[t]

new+=k[j]

except IndexError:

pass

new+=" "

return new

if \_\_name\_\_ == '\_\_main\_\_':

fptr = open(os.environ['OUTPUT\_PATH'], 'w')

s = input()

result = encryption(s)

fptr.write(result + '\n')

fptr.close()

1. Maximizing XOR

#!/bin/python3

import math

import os

import random

import re

import sys

def maximizingXor(l, r):

    comp=0

    for j in range(l,r+1):

        for k in range(l,r+1):

            if j<=k:

                a=bin(j)

                b=bin(k)

                c=a.replace("0b","")

                d=b.replace("0b","")

                p=32-len(c)

                q=32-len(d)

                c=list(c)

                d=list(d)

                for i in range(0,p):

                    c.insert(0,'0')

                for i in range(0,q):

                    d.insert(0,'0')

                m=len(c)

                n=len(d)

                s=""

                for i in range(0,m):

                    if c[i]=='1' and d[i]=='0':

                        s+='1'

                    elif c[i]=='0' and d[i]=='1':

                        s+='1'

                    else:

                        s+='0'

                x=int(s,2)

                # print(j,"+",k,"=",x," and ",c,"+",d,"=",s)

                if comp<x:

                    comp=x

                else:

                    pass

            else:

                pass

    return comp

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    l = int(input().strip())

    r = int(input().strip())

    result = maximizingXor(l, r)

    fptr.write(str(result) + '\n')

    fptr.close()

1. QuickSort 1-Partition

#!/bin/python3

import math

import os

import random

import re

import sys

def quickSort(arr):

    s=[]

    r=[]

    le=len(arr)

    m=round(le/2)

    p=arr[0]

    for i in range(1,le):

        if arr[i]>p:

            r.append(arr[i])

        elif arr[i]<p:

            s.append(arr[i])

    s.append(p)

    for j in range(0,m):

        s.append(r[j])

    return s

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    n = int(input().strip())

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

    result = quickSort(arr)

    fptr.write(' '.join(map(str, result)))

    fptr.write('\n')

    fptr.close()

1. Append and Delete

#!/bin/python3

import math

import os

import random

import re

import sys

def appendAndDelete(s, t, k):

    c=0

    for i,j in zip(s,t):

            if i==j:

                c+=1

            else:

                break

    l=len(s)+len(t)

    if l<= 2\*c+k and l%2==k%2 or l<k:

        return 'Yes'

    else:

        return 'No'

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    s = input()

    t = input()

    k = int(input().strip())

    result = appendAndDelete(s, t, k)

    fptr.write(result + '\n')

    fptr.close()

1. Almost Sorted

#!/bin/python3

import math

import os

import random

import re

import sys

import copy

def almostSorted(arr):

    # Write your code here

    s=copy.deepcopy(arr)

    s.sort()

    if s==arr:

        print("yes")

        return

    a=b=-1

    for i in range(n-1):

        if arr[i]>arr[i+1]:

            a=i

            break

    for i in range(n-1,0,-1):

        if arr[i]<arr[i-1]:

            b=i

            break

    t=copy.deepcopy(arr)

    t[a],t[b]=t[b],t[a]

    if t == s:

        print("yes")

        print("swap",a+1,b+1)

        return

    t=copy.deepcopy(arr)

    t = t[:a] + t[a:b+1][::-1] + t[b+1:]

    if t == s:

        print("yes")

        print("reverse",a+1,b+1)

        return

    print("no")

if \_\_name\_\_ == '\_\_main\_\_':

    n = int(input().strip())

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

    almostSorted(arr)

1. Quick Sort 1

#!/bin/python3

import math

import os

import random

import re

import sys

def quickSort(arr):

    l=[]

    r=[]

    s=[]

    le=len(arr)

    p=arr[0]

    for i in range(1,le):

        if arr[i]>p:

            r.append(arr[i])

        elif arr[i]<p:

            l.append(arr[i])

    li=len(l)

    ri=len(r)

    for i in range(0,li):

        s.append(l[i])

    s.append(p)

    for j in range(0,ri):

        s.append(r[j])

    return s

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    n = int(input().strip())

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

    result = quickSort(arr)

    fptr.write(' '.join(map(str, result)))

    fptr.write('\n')

    fptr.close()

1. **Sub array**

import math

import os

import random

import re

import sys

def birthday(s, d, m):

    c=0

    if len(s)>1:

        for i in range(0,len(s)):

            add=s[0]

            for j in range(1,m):

                try:

                    add+=s[j]

                    if add==d and j==m-1:

                        c+=1

                except IndexError:

                    pass

            s.pop(0)

    elif s[0]==d:

        c+=1

    return c

if \_\_name\_\_ == '\_\_main\_\_':

    fptr = open(os.environ['OUTPUT\_PATH'], 'w')

    n = int(input().strip())

    s = list(map(int, input().rstrip().split()))

    first\_multiple\_input = input().rstrip().split()

    d = int(first\_multiple\_input[0])

    m = int(first\_multiple\_input[1])

    result = birthday(s, d, m)

    fptr.write(str(result) + '\n')

    fptr.close()