

```
In [ ]: #Introductions
if __name__ == '__main__':
    print("Hello, World!")
```

```
In [ ]: if __name__ == '__main__':
        n = int(input().strip())
        if n % 2 == 1:
            print("Weird")
        elif n % 2 == 0 and 2 <= n <= 5:
            print("Not Weird")
        elif n % 2 == 0 and 6 <= n <= 20:
            print("Weird")
        else:
            print("Not Weird")
```

```
In [ ]: if __name__ == '__main__':
        a = int(input())
        b = int(input())
        print (a+b)
        print (a-b)
        print (a*b)
```

```
In [ ]: if __name__ == '__main__':
        a = int(input())
        b = int(input())
        print(a//b)
        print(a/b)
```

```
In [ ]: if __name__ == '__main__':
        n = int(input())
        for i in range (n) :
            print (i**2)
```

```
In [ ]: def is_leap(year):
        leap = False
        if year%4==0:
            leap = True
            if year%100==0:
                leap = False
                if year%400==0:
                    leap = True
        return leap
        if leap == True:
            return True
        else:
            return False
```

```
In [ ]: if __name__ == '__main__':
        n = int(input())
        for i in range (1,n+1):
            print (i, end='')
```

```
In [ ]: #Basic data types
if __name__ == '__main__':
    x = int(input())
    y = int(input())
    z = int(input())
    n = int(input())
```

```
a=[[i,j,k] for i in range (0,x+1) for j in range (0,y+1) for k in range (0,z+1)]
print(a)
```

```
In [ ]: if __name__ == '__main__':
        n = int(input())
        arr = map(int, input().split())

        l=list(arr)
        a=[]
        for i in l:
            if i not in a:
                a.append(i)
        a.remove(max(a))
        print(max(a))
```

```
In [ ]: Result =[]
        scorelist = []
        if __name__ == '__main__':
            for _ in range(int(input())):
                name = input()
                score = float(input())
                Result+=[[name,score]]
                scorelist+= [score]
            b=sorted(list(set(scorelist)))[1]
            for a,c in sorted(Result):
                if c==b:
                    print(a)
```

```
In [ ]: if __name__ == '__main__':
        n = int(input())
        student_marks = {}
        for _ in range(n):
            name, *line = input().split()
            scores = list(map(float, line))
            student_marks[name] = scores
        query_name = input()

        l= list(student_marks[query_name])
        m = sum(l)/len(l)
        print("%.2f" % m)
```

```
In [ ]: if __name__ == '__main__':
        n = int(input())
        A=[]
        for i in range(n):
            A.append(input().split())
        B=[]
        for i in range(n):
            if A[i][0]=='insert':
                B.insert(int(A[i][1]),int(A[i][2]))
            elif A[i][0]=='print':
                print(B)
            elif A[i][0]=='remove':
                B.remove(int(A[i][1]))
            elif A[i][0]=='append':
                B.append(int(A[i][1]))
```

```
In [ ]: #Strings
        def print_full_name(first, last):
            print("Hello " + first, last + "! You just delved into python.")
```

```
In [ ]: #Sets
```

```
def average(array):
    array = set(array)
    return sum(array) / len(array)
```

```
In [ ]: inp=input().split()
n=int(inp[0])
m=int(inp[1])
N=list(map(int, input().strip().split()))
A=set(map(int, input().strip().split()))
B=set(map(int, input().strip().split()))
happiness=0
for i in N:
    if i in A:
        happiness+=1
    elif i in B:
        happiness=happiness-1
    else:
        happiness=happiness
print(happiness)
```

```
In [ ]: M = int(input())
setm = set(map(int, input().split()))
N = int(input())
setn = set(map(int, input().split()))

defm = setm.difference(setn)
defn = setn.difference(setm)

diff = defm.union(defn)

for i in sorted(list(diff)):
    print(i)
```

```
In [ ]: N = int(input())
countries = set()

for i in range(N):
    countries.add(input())

print(len(countries))
```

```
In [ ]: n=int(input())
N=set(map(int, input().split()))
b=int(input())
B=set(map(int, input().split()))
s=N.union(B)
print(len(s))
```

```
In [ ]: n=int(input())
N=set(map(int, input().split()))
b=int(input())
B=set(map(int, input().split()))
s=N.intersection(B)
print(len(s))
```

```
In [ ]: n=int(input())
N=set(map(int, input().split()))
b=int(input())
B=set(map(int, input().split()))
s=N.difference(B)
print(len(s))
```

```
In [ ]: n=int(input())
N=set(map(int, input().split()))
b=int(input())
B=set(map(int, input().split()))
s=N.symmetric_difference(B)
print(len(s))
```

```
In [ ]: K=int(input())
l=map(int, input().split())
l= sorted(l)

for i in range(len(l)):
    if(i != len(l)-1):
        if(l[i]!=l[i-1] and l[i]!=l[i+1]):
            print(l[i])
            break;
    else:
        print(l[i])
```

```
In [ ]: #Date and time
import datetime
import calendar
m, d, y = map(int, input().split())
input_date = datetime.date(y, m, d)
print(calendar.day_name[input_date.weekday()].upper())
```

```
In [ ]: #Exceptions
T = int(input())
for i in range(T):
    try:
        a, b = map(int, input().split())
        print(a//b)
    except Exception as e:
        print("Error Code:",e)
```

```
In [ ]: #Built-ins
N,X=input().split()
l=list()
for i in range (int(X)):
    mark=map(float, input().split())
    l.append(mark)
for j in zip(*l):
    print(sum(j)/len(j))
```

```
In [ ]: import math
import os
import random
import re
import sys
if __name__ == '__main__':
    n,m = map(int, input().split())
    arr = []
    for _ in range(n):
        arr.append(list(map(int, input().rstrip().split())))
    k = int(input())
    arr.sort(key = lambda x : x[k])
    for i in arr:
        print(*i,sep=' ')
```

```
In [ ]: print(*sorted(input(), key=lambda c: (c.isdigit() - c.islower(), c in '02468', c)),
```

```
In [ ]: #Map and Lambda Expressions
cube = lambda x: x**3

def fibonacci(n):
    l = [0, 1]
    for i in range(2, n):
        l.append(l[i-1] + l[i-2])
    return(l[0:n])
```

```
In [ ]: #Decorators
def wrapper(f):
    def fun(l):
        f(['+91 ' + i[-10:-5] + ' ' + i[-5:] for i in l])
    return fun
```

```
In [ ]: def person_lister(f):
    def inner(people):
        return map(f, sorted(people, key=lambda x: int(x[2])))
    return inner
```

```
In [ ]: #Numpy
def arrays(arr):
    return numpy.array(arr[::-1],float)
```

```
In [ ]: import numpy
l=list(map(int,input().split()))
arr=numpy.array(l)
print(numpy.reshape(arr,(3,3)))
```

```
In [ ]: import numpy

N,M=map(int,input().split())
a=[]
for i in range(N):
    l=list(map(int,input().split()))
    a.append(l)
arr=numpy.array(a)
print(numpy.transpose(arr))
print(arr.flatten())
```

```
In [ ]: import numpy as np
N,M,P=map(int, input().split())
a=[]
b=[]
for i in range(N):
    l=list(map(int, input().split()))
    a.append(l)
for j in range(M):
    m=list(map(int, input().split()))
    b.append(m)
print(np.concatenate((a,b)))
```

```
In [ ]: import numpy as np
N = tuple(map(int, input().split()))
print(np.zeros(N, int))
print(np.ones(N, int))
```

```
In [ ]: import numpy as np
np.set_printoptions(sign=' ')
print(np.eye(*map(int, input().split())))
```

```
In [ ]: import numpy as np
N,M=map(int, input().split())
A=[]
B=[]
for i in range (N):
    l=list(map(int, input().split()))
    A.append(l)
for j in range(N):
    m=list(map(int, input().split()))
    B.append(m)
print(np.add(A,B))
print(np.subtract(A,B))
print(np.multiply(A,B))
print(np.floor_divide(A,B))
print(np.mod(A,B))
print(np.power(A,B))
```

```
In [ ]: import numpy as np
np.set_printoptions(legacy='1.13')
A=np.array(input().split(), float)
print(np.floor(A))
print(np.ceil(A))
print(np rint(A))
```

```
In [ ]: import numpy as np
N,M=map(int, input().split())
A=[]
for i in range(N):
    l=list(map(int, input().split()))
    A.append(l)
    S=np.sum(A, axis=0)
print(np.prod(S))
```

```
In [ ]: import numpy as np
N,M=map(int, input().split())
A=[]
for i in range(N):
    l=list(map(int, input().split()))
    A.append(l)
    minimo=np.min(A, axis=1)
print(np.max(minimo))
```

```
In [ ]: import numpy as np
N,M=map(int, input().split())
A=[]
for i in range(N):
    l=list(map(int, input().split()))
    A.append(l)
print(np.mean(A, axis=1))
print(np.var(A, axis=0))
print(round(np.std(A, axis=None), 11))
```

```
In [ ]: import numpy as np
N=int(input())
A=[]
B=[]
for i in range(N):
    l=list(map(int, input().split()))
    A.append(l)
for j in range(N):
    m=list(map(int, input().split()))
```

```
B.append(m)
print(np.dot(A,B))
```

```
In [ ]: import numpy as np
A=np.array(input().split(), int)
B=np.array(input().split(), int)
print(np.inner(A,B))
print(np.outer(A,B))
```

```
In [ ]: import numpy as np
P=list(map(float,input().split()))
x=float(input())
print(np.polyval(P,x))
```

```
In [ ]: import numpy as np
N=int(input())
A=[]
for i in range(N):
    l=list(map(float, input().split()))
    A.append(l)
print(round(np.linalg.det(A),2))
```

```
In [ ]: #Candles
import math
import os
import random
import re
import sys
def birthdayCakeCandles(candles):
    i=0
    m=max(candles)
    for j in candles:
        if j==m:
            i+=1
    return i
if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    candles_count = int(input().strip())

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

    result = birthdayCakeCandles(candles)

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

    fptr.close()
```

```
In [ ]: #kangaroo
import math
import os
import random
import re
import sys
def kangaroo(x1, v1, x2, v2):
    if x2>x1 and v2>v1:
        return "NO"
    else:
        if v2-v1==0:
            return 'NO'
        else:
            result=(x1-x2)%(v2-v1)
```

```

        if result==0:
            return 'YES'
        else:
            return 'NO'

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    first_multiple_input = input().rstrip().split()

    x1 = int(first_multiple_input[0])

    v1 = int(first_multiple_input[1])

    x2 = int(first_multiple_input[2])

    v2 = int(first_multiple_input[3])

    result = kangaroo(x1, v1, x2, v2)

    fptr.write(result + '\n')

    fptr.close()

```

```

In [ ]: #Adv
import math
import os
import random
import re
import sys
def viralAdvertising(n):
    s=5
    a=0
    for i in range(1,n+1):
        l=s//2
        a=a+1
        s=l*3
    return a

if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    n = int(input().strip())

    result = viralAdvertising(n)

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

    fptr.close()

```

In []:

In []: