

# Homework\_1

October 8, 2022

## 1 Say “Hello, World!” With Python

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

## 2 List Comprehensions

```
[ ]: if __name__ == '__main__':  
    x = int(input())  
    y = int(input())  
    z = int(input())  
    n = int(input())  
    l=[]  
    for i in range(0, x+1):  
        for j in range(0, y+1):  
            for k in range(0, z+1):  
                if i+j+k==n:  
                    pass  
                else:  
                    l.append([i, j, k])  
  
    print(l)
```

## 3 sWAP cASE

```
[ ]: def swap_case(s):  
    c=''  
    for i in s:  
        c+=i.swapcase()  
    return c
```

## 4 Python If-Else

```
[ ]: #!/bin/python3

import math
import os
import random
import re
import sys

if __name__ == '__main__':
    n = int(input().strip())
    if n%2!=0:
        print('Weird')
    if n%2==0:
        if n>=2 and n<=5:
            print('Not Weird')
        if n>=6 and n <= 20:
            print ('Weird')
        if n>20:
            print ('Not Weird')
```

## 5 Arithmetic Operators

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

    somma=a+b
    print(somma)
    diff=a-b
    print(diff)
    molt=a*b
    print(molt)
```

## 6 Python: Division

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

    intera=a//b
    print(intera)
    div=a/b
```

```
print(div)
```

## 7 Loops

```
[ ]: if __name__ == '__main__':  
    n = int(input())  
  
    for i in range (0, n):  
        print(i**2)
```

## 8 Write a function

```
[ ]: def is_leap(year):  
    leap = False  
    if year%4==0:  
        leap = True  
        if year%100==0:  
            leap=False  
            if year%400==0:  
                leap=True  
  
    # Write your logic here  
  
    return leap
```

## 9 Print Function

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

## 10 String Split and Join

```
[ ]: def split_and_join(line):  
    line=line.replace(' ', '-')  
    return line  
  
if __name__ == '__main__':  
    line = input()
```

```
result = split_and_join(line)
print(result)
```

## 11 What's Your Name?

```
[ ]: #
# Complete the 'print_full_name' function below.
#
# The function is expected to return a STRING.
# The function accepts following parameters:
# 1. STRING first
# 2. STRING last
#

def print_full_name(first, last):
    # Write your code here
    print('Hello ' + first + ' ' + last + '! You just delved into python.')
```

## 12 Mutations

```
[ ]: def mutate_string(string, position, character):
    l=list(string)
    l[position]=character
    string=''.join(l)
    return string
```

## 13 Find the Runner-Up Score!

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

    l=[]
    for i in arr:
        l.append(i)

    l=list(dict.fromkeys(l))
    l.sort(reverse=True)
    print(l[1])
```

## 14 Find a string

```
[ ]: def count_substring(string, sub_string):
    conta=0
    if len(string) in range (1,201):
        for i in range (0, len(string)):
            if string[i]==sub_string[0]:
                lettere=0
                for j in range(0, len(sub_string)):
                    if i+j>=len(string):
                        return conta
                    if string[i+j]==sub_string[j]:
                        lettere+=1
                    if lettere==len(sub_string):
                        lettere=0
                        conta+=1

    return conta
```

## 15 String Validators

```
[ ]: if __name__ == '__main__':
    s = input()

    if len(s) in range(1, 1001):
        print(any(i.isalnum() for i in s))

    if len(s) in range(1, 1001):
        print(any(i.isalpha() for i in s))

    if len(s) in range(1, 1001):
        print(any(i.isdigit() for i in s))

    if len(s) in range(1, 1001):
        print(any(i.islower() for i in s))

    if len(s) in range(1, 1001):
        print(any(i.isupper() for i in s))
```

## 16 List Comprehensions

```
[ ]: if __name__ == '__main__':
    x = int(input())
    y = int(input())
    z = int(input())
    n = int(input())
```

```

mix=[[i,j,k] for i in range(x+1) for j in range (y+1) for k in range (z+1) if
↪ i+j+k!=n]
print (mix)

```

## 17 Nested Lists

```

[ ]: if __name__ == '__main__':
    l=[]
    for _ in range(int(input())):
        name = input()
        score = float(input())
        l.append([name, score])

    x=[]
    l=sorted(l, key = lambda x: x[1])
    for i in range(0, len(l)):
        if l[i][1]!=l[0][1]:
            x.append(l[i])
    ordinata=[]
    ordinata.append(x[0])
    if len(x)>0:
        for i in range(1,len(x)):
            if x[i][1]==x[0][1]:
                ordinata.append(x[i])

    ordinata.sort()
    for i in range(0, len(ordinata)):
        print(ordinata[i][0])

```

## 18 Finding the percentage

```

[ ]: 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()

    dizionario={}
    for i in student_marks:
        dizionario[i]=sum(student_marks[i])/len(student_marks[i])

    print("{:.2f}".format(dizionario[query_name]))

```

## 19 Lists

```
[ ]: if __name__ == '__main__':
    N = int(input())
    comandi=[]
    for i in range(N):
        comandi.append(input().split())

    l=[]
    for i in range(N):
        if comandi[i][0]=='insert':
            l.insert(int(comandi[i][1]), int(comandi[i][2]))
        elif comandi[i][0]=='print':
            print(l)
        elif comandi[i][0]=='remove':
            l.remove(int(comandi[i][1]))
        elif comandi[i][0]=='append':
            l.append(int(comandi[i][1]))
        elif comandi[i][0]=='pop':
            l.pop()
        elif comandi[i][0]=='sort':
            l.sort()
        elif comandi[i][0]=='reverse':
            l.reverse()
```

## 20 Tuples

```
[ ]: if __name__ == '__main__':
    n = int(input())
    integer_list = map(int, input().split())
    t=tuple(integer_list)
    print(hash(t))
```

## 21 Text Alignment

```
[ ]: #Replace all _____ with rjust, ljust or center.

thickness = int(input()) #This must be an odd number
c = 'H'

#Top Cone
for i in range(thickness):
    print((c*i).rjust(thickness-1)+c+(c*i).ljust(thickness-1))

#Top Pillars
for i in range(thickness+1):
```

```

    print((c*thickness).center(thickness*2)+(c*thickness).center(thickness*6))

#Middle Belt
for i in range((thickness+1)//2):
    print((c*thickness*5).center(thickness*6))

#Bottom Pillars
for i in range(thickness+1):
    print((c*thickness).center(thickness*2)+(c*thickness).center(thickness*6))

#Bottom Cone
for i in range(thickness):
    print(((c*(thickness-i-1)).rjust(thickness)+c+(c*(thickness-i-1)).
    ↪ljust(thickness)).rjust(thickness*6))

```

## 22 Text Wrap

```

[ ]: def wrap(string, max_width):
    for i in range(0, len(string)+1, max_width):
        line=string[i:i+max_width]
        if len(line)==max_width:
            print (line)
        else:
            return line

```

## 23 Designer Door Mat

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
N, M = map(int, input().split())
for i in range (1, N, 2):
    print(str('.|.' * i).center(M, '-'))
print('WELCOME'.center(M, '-'))
for i in range(N-2, -1, -2):
    print(str('.|.' * i).center(M, '-'))

```

## 24 String Formatting

```

[ ]: def print_formatted(number):
    if number >=1 and number <=99:
        width = len(bin(number)[2:])
        for i in range(1,number+1):
            print(str(i).rjust(width, ' '),end=" ")
            print(oct(i)[2:].rjust(width, ' '),end=" ")
            print(((hex(i)[2:]).upper()).rjust(width, ' '),end=" ")
            print(bin(i)[2:].rjust(width, ' '),end=" ")

```



```
print('')

# your code goes here
```

## 25 Alphabet Rangoli

```
[ ]: def print_rangoli(size):
    # your code goes here
    alfabeto = 'abcdefghijklmnopqrstuvwxyz'
    riga = []
    for i in range(size):
        r = "-".join(alfabeto[i:size])
        riga.append((r[::-1]+r[1:]).center(4*size-3, "-"))

    print('\n'.join(riga[:0:-1]+riga))
```

## 26 Capitalize!

```
[ ]: # Complete the solve function below.
def solve(s):
    for i in s.split():
        s = s.replace(i, i.capitalize())
    return s
```

## 27 The Minion Game

```
[ ]: def minion_game(string):
    l=len(string)
    kevin = 0
    stuart = 0
    for i in range(l):
        if string[i] in 'AEIOU':
            kevin+=(l-i)
        else:
            stuart+=(l-i)
    if kevin < stuart:
        print('Stuart ' + str(stuart))
    elif kevin > stuart:
        print('Kevin ' + str(kevin))
    else:
        print('Draw')
```

## 28 Merge the Tools!

```
[ ]: def merge_the_tools(string, k):
    if len(string)%k==0:
        l=[]
        a=[]
        parola=''
        for i in string:
            parola+=i
            if len(parola)==k:
                a.append(parola)
                l.append(a)
                parola=''
                a=[]
        for j in l:
            conta=0
            parola2=''
            for k in j:
                conta+=1
                for x in k:
                    if x not in parola2:
                        parola2+=x
            if conta==len(j):
                print(parola2)
```

## 29 Introduction to Sets

```
[ ]: def average(array):
    if len(set(arr))>0 and len(set(arr))<=100:
        s=sum(set(arr))
        d=len(set(arr))
        return(round(s/d, 3))
```

## 30 Symmetric Difference

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
M=int(input())
a = set(map(int, input().split()))
N = int(input())
b = set(map(int, input().split()))
adiff = a.difference(b)
bdiff = b.difference(a)

unione=adiff.union(bdiff)

for i in sorted(list(unione)):
```

```
print(i)
```

## 31 No Idea!

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
numeri=input().split()
m = int(numeri[0])
n = int(numeri[1])
happiness = 0
arr = list(map(int, input().split()))
A = set(map(int, input().split()))
B = set(map(int, input().split()))

if n>=1 and n<=10**5:
    if m>=1 and m <=10**5:
        for i in arr:
            if i in A:
                happiness+=1
            if i in B:
                happiness-=1

print(happiness)
```

## 32 Set .add()

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
n=int(input())
nazioni=set()
for i in range(n):
    nazioni.add(input())

print(len(nazioni))
```

## 33 Set .discard(), .remove() & .pop()

```
[ ]: n = int(input())
s = set(map(int, input().split()))
N=int(input())

if n>0 and n<20:
    if N>0 and N<20:
        for i in range (N):
            comando=input().split()
            if comando[0]=='remove':
                s.remove(int(comando[1]))
```

```

        elif comando[0]=='discard':
            s.discard(int(comando[1]))
        else:
            s.pop()

print(sum(s))

```

### 34 Set .union() Operation

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT

neng=int(input())
eng=set(input().split())

nfr=int(input())
fr=set(input().split())

unione=fr.union(eng)

print(len(unione))

```

### 35 Set .intersection() Operation

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT

neng=int(input())
eng=set(input().split())

nfr=int(input())
fr=set(input().split())

intersezione=fr.intersection(eng)

print(len(intersezione))

```

### 36 Set .difference() Operation

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT

neng=int(input())
eng=set(input().split())

nfr=int(input())
fr=set(input().split())

differenza=eng.difference(fr)

```

```
print(len(differenza))
```

## 37 Set .symmetric\_difference() Operation

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
neng=int(input())
eng=set(input().split())

nfr=int(input())
fr=set(input().split())

simdif=eng.symmetric_difference(fr)

print(len(simdif))
```

## 38 Set Mutations

```
[ ]: NA=int(input())
A=set(map(int, input().split()))
N=int(input())
if len(A)<1000 and len(A)>0:
    if N>0 and N<100:
        for i in range(N):
            operazioni, i = input().split(' ')
            b = set(map(int, input().split(' ')))
            if operazioni == "update":
                A.update(b)
            elif operazioni == "intersection_update":
                A.intersection_update(b)
            elif operazioni == "difference_update":
                A.difference_update(b)
            elif operazioni == "symmetric_difference_update":
                A.symmetric_difference_update(b)
print(sum(A))
```

## 39 The Captain's Room

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
K=int(input())
stanze=map(int, input().split())
dizstanze={}
if K>1 and K<1000:
    for i in stanze:
        dizstanze[i]=dizstanze.get(i, 0)+ 1
```

```

for k,v in dizstanze.items():
    if v==1:
        print(k)

```

## 40 Check Subset

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
T=int(input())
if T >0 and T<21:
    for i in range(T):
        nA=int(input())
        A=set(input().split())
        nB=int(input())
        B=set(input().split())
        if nA>0 and nA<1001 and nB>0 and nB<1001:
            if A.intersection(B)==A:
                print('True')
            else:
                print('False')

```

## 41 Check Strict Superset

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
A=set(map(int, input().split()))
n=int(input())
if len(A)>0 and len(A)<501 and n>0 and n<21:
    conta=0
    for i in range(n):
        s=set(map(int, input().split()))
        if len(s)<101 and len(s)>0:
            if A.intersection(s)==s and len(A.intersection(s))<len(A):

                conta+=1
    if conta==n:
        print('True')
    else:
        print('False')

```

## 42 collections.Counter()

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
from collections import Counter

X=int(input())
scarpe=Counter(map(int, input().split()))

```

```

N=int(input())
guadagni=0
if X<10**3 and X>0 and N>0 and N<=10**3:
    for i in range(N):
        taglie,value=map(int, input().split())
        # if value<100 and value>20:
        if scarpe[taglie]>0:
            scarpe[taglie]-=1
            guadagni+=value
print(guadagni)

```

## 43 DefaultDict Tutorial

[ ]: *# Enter your code here. Read input from STDIN. Print output to STDOUT*

```

from collections import defaultdict
A=defaultdict(list)
n, m= map(int, input().split())

if n>=1 and n<=10000:
    if m>=1 and m<=100:
        for i in range (1,n+1):
            A[input()].append(str(i))

        for i in range(m):
            parola=input()
            if parola in A:
                for j in A:
                    if j==parola:
                        print (' '.join(A[j]))
            else:
                print(-1)

```

## 44 Collections.namedtuple()

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
from collections import namedtuple
N=int(input())
variabili=input().split()
somma=0

if N>0 and N<=100:
    for i in range(N):
        somma+=int(input().split()[variabili.index("MARKS")])

print(round(float(somma/N),2))
```

## 45 Arrays

```
[ ]: def arrays(arr):
    arr.reverse()
    return numpy.array(arr, float)
```

## 46 Collections.OrderedDict()

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
from collections import OrderedDict

N=int(input())
ordered_dictionary=OrderedDict()
if N>0 and N<=100:
    for i in range(N):
        item_name, net_price=input().rsplit(' ',1)
        if item_name in ordered_dictionary:
            ↵
            ↪ordered_dictionary[item_name]=ordered_dictionary[item_name]+int(net_price)
        else:
            ordered_dictionary[item_name]=int(net_price)

    for i,j in ordered_dictionary.items():
        print(i,j)
```

## 47 Min and Max

```
[ ]: import numpy

N,M=map(int, input().split())
array=numpy.array([list(map(int, input().split())) for i in range(N)])
```



```
print(numpy.max(numpy.min(array, axis=1)))
```

## 48 Shape and Reshape

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

## 49 Word Order

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
n=int(input())
d={}
if n>=1 and n<=10**5:
    for i in range(n):
        parola=input()
        if parola in d:
            d[parola]+=1
        else:
            d[parola]=1

print(len(d))
for i in d.items():
    print(i[1], end=' ')
```

## 50 Collections.deque()

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
from collections import deque

N=int(input())
d=deque()
if N>0 and N<=100:
    for i in range(N):
        comando=input().split()
        if 'append' in comando:
            d.append(comando[1])
        elif 'pop' in comando:
            d.pop()
        elif 'popleft' in comando:
            d.popleft()
        elif 'appendleft' in comando:
            d.appendleft(comando[1])

for i in d:
```

```
print(i, end=' ')
```

## 51 Company Logo

```
[ ]: #!/bin/python3

import math
import os
import random
import re
import sys

if __name__ == '__main__':
    s = input()
    d={}

    if len(s)>3 and len(s)<=10**4:
        for i in s:
            if i in d:
                d[i]+=1
            else:
                d[i]=1

        val_ord=sorted(d.values(), reverse=True)
        key_ord=sorted(d.keys())
        ordinato={}
        for i in val_ord:
            for j in key_ord:
                if d[j]==i:
                    ordinato[j]=i

        conta=0
        for i, j in ordinato.items():
            print(i, j)
            conta+=1
            if conta==3:
                break
```

## 52 Piling Up!

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
from collections import deque
T=int(input())
righe=[]
```

```

if T>=1 and T<=5:
    for i in range(T):
        n=int(input())
        if n>=1 and n<=10**5:
            righe.append(deque(list(map(int, input().split()))))

for i in righe:
    stack=[]
    if i[0]>=i[-1]:
        stack.append(i.popleft())
    else:
        stack.append(i.pop())
    while len(i)>0:
        left=i[0]
        right=i[-1]
        top=stack[-1]
        if left<=top and left >=right:
            stack.append(i.popleft())
        elif right<=top and right>=left:
            stack.append(i.pop())
        else:
            break
    if len(i)==0:
        print('Yes')
    else:
        print('No')

```

## 53 Calendar Module

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT

import calendar
m, d, y=list(map(int, input().split()))

if y>2000 and y<3000:
    print(calendar.day_name[calendar.weekday(y,m,d)].upper())

```

## 54 Time Delta

```

[ ]: #!/bin/python3

import math
import os
import random
import re
import sys

```

```

from datetime import datetime
# Complete the time_delta function below.
def time_delta(t1, t2):
    dt1=datetime.strptime(t1, '%a %d %b %Y %H:%M:%S %z')
    dt2=datetime.strptime(t2, '%a %d %b %Y %H:%M:%S %z')
    d=abs(dt1-dt2)
    return (str(int(d.total_seconds()))))

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

    t = int(input())

    for t_itr in range(t):
        t1 = input()

        t2 = input()

        delta = time_delta(t1, t2)

        fptr.write(delta + '\n')

    fptr.close()

```

## 55 Exceptions

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
T= int(input())

if T>0 and T<10:
    for i in range(T):
        try:
            a, b=map(int, input().split())
            print(a//b)
        except(ValueError, ZeroDivisionError) as e:
            print('Error Code:', e)

```

## 56 Zipped!

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
N, X=map(int, input().split())
voti=[]
if N>0 and N<=100 and X>0 and X<=100:
    for i in range(X):
        voti.append(list(map(float, input().split())))

```

```

for j in range(N):
    s=0
    for k in range(X):
        s+=voti[k][j]

    print(round(s/X,1))

```

## 57 Athlete Sort

```

[ ]: #!/bin/python3

import math
import os
import random
import re
import sys

if __name__ == '__main__':
    nm = input().split()

    n = int(nm[0])

    m = int(nm[1])

    arr = []

    for _ in range(n):
        arr.append(list(map(int, input().rstrip().split())))

    k = int(input())

    if n>=1 and n<=1000 and m>=1 and m<=1000:
        for i in sorted(arr, key=lambda x:x[k]):
            print(*i)

```

## 58 ginortS

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
stringa=input()
l,u='',''
o,e='',''
for i in stringa:
    if i.islower():
        l+=i

```

```

elif i.isupper():
    u+=i
elif i.isdigit() and int(i)%2!=0:
    o+=i
else:
    e+=i

s=sorted(l)+sorted(u)+sorted(o)+sorted(e)
print(''.join(s))

```

## 59 Map and Lambda Function

```

[ ]: cube = lambda x: x**3

def fibonacci(n):
    if n>=0 and n<=15:
        a,b=0, 1
        l=[]
        for i in range (n):
            l.append(a)
            a, b= b, a+b
        return l

# return a list of fibonacci numbers

```

## 60 XML 1 - Find the Score

```

[ ]: def get_attr_number(node):
    s=0
    for i in node.iter():
        a=i.attrib
        s+=len(a)
    return s

```

## 61 XML2 - Find the Maximum Depth

```

[ ]: maxdepth = 0
def depth(elem, level):
    global maxdepth
    for i in elem:
        depth(i, level+1)
    maxdepth=max(level+1, maxdepth)

```

## 62 Standardize Mobile Number Using Decorators

```
[ ]: def wrapper(f):  
    def fun(l):  
        l1=['+91 '+i[-10:-5]+' '+ i[-5:]]  
        f(l1)  
    return fun
```

## 63 Decorators 2 - Name Directory

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

## 64 Polynomials

```
[ ]: import numpy  
  
poly= list(map(float, input().split()))  
p=float(input())  
  
print(numpy.polyval(poly, p))
```

## 65 Linear Algebra

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

## 66 Transpose and Flatten

```
[ ]: import numpy

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

## 67 Concatenate

```
[ ]: import numpy

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

## 68 Zeros and Ones

```
[ ]: import numpy

dim=list(map(int, input().split()))
print(numpy.zeros(dim, int))
print(numpy.ones(dim, int))
```

## 69 Eye and Identity

```
[ ]: import numpy
numpy.set_printoptions(legacy='1.13')

N, M=map(int, input().split())
```



```
print(numpy.eye(N,M))
```

## 70 Array Mathematics

```
[ ]: import numpy

N, M=map(int, input().split())
A=[]
B=[]
for i in range(N):
    A.append(input().split())
A=numpy.array(A, int)
for i in range(N):
    B.append(input().split())
B=numpy.array(B, int)
print(A+B)
print(A-B)
print(A*B)
print(A//B)
print(A%B)
print(A**B)
```

## 71 Floor, Ceil and Rint

```
[ ]: import numpy
numpy.set_printoptions(legacy='1.13')

arr=list(map(float, input().split()))
arr=numpy.array(arr)
print(numpy.floor(arr))
print(numpy.ceil(arr))
print(numpy rint(arr))
```

## 72 Sum and Prod

```
[ ]: import numpy

N, M= input().split()
l=[]
for i in range(int(N)):
    l.append(input().split())
```

```
l=numpy.array(l, int)
s=numpy.sum(l, axis=0)
print(numpy.prod(s))
```

## 73 Mean, Var, and Std

```
[ ]: import numpy

N, M=list(map(int, input().split()))
l=numpy.array([input().split() for i in range(N)], int)
print(numpy.mean(l, axis=1))
print(numpy.var(l, axis=0))
print(round(numpy.std(l, axis=None), 11))
```

## 74 Dot and Cross

```
[ ]: import numpy

N=int(input())
A=[]
B=[]
for i in range(N):
    A.append(numpy.array(input().split(), int))

for i in range(N):
    B.append(numpy.array(input().split(), int))

print(numpy.dot(A,B))
```

## 75 Inner and Outer

```
[ ]: import numpy

A=numpy.array(input().split(), int)
B=numpy.array(input().split(), int)
print(numpy.inner(A, B))
print(numpy.outer(A,B))
```

## 76 Birthday Cake Candles

```
[ ]: #!/bin/python3

import math
import os
import random
import re
import sys
from collections import Counter
#
# Complete the 'birthdayCakeCandles' function below.
#
# The function is expected to return an INTEGER.
# The function accepts INTEGER_ARRAY candles as parameter.
#

def birthdayCakeCandles(candles):
    b=0
    a=max(candles)
    for i in candles:
        if a==i:
            b+=1
    return b
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()
```

## 77 Number Line Jumps

```
[ ]: #!/bin/python3

import math
import os
import random
import re
import sys
```

```

#
# Complete the 'kangaroo' function below.
#
# The function is expected to return a STRING.
# The function accepts following parameters:
# 1. INTEGER x1
# 2. INTEGER v1
# 3. INTEGER x2
# 4. INTEGER v2
#

def kangaroo(x1, v1, x2, v2):
    if x1<=x2 and x1>=0 and x2<=10000:
        if v1>=1 and v1<=10000 and v2<=10000 and v1>=1:
            c=x2-x1
            for i in range(c):
                x1=x1+v1
                x2=x2+v2
                if x1==x2:
                    return 'YES'
            return 'NO'
    # Write your code here

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()

```

## 78 Viral Advertising

```
[ ]: #!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'viralAdvertising' function below.
#
# The function is expected to return an INTEGER.
# The function accepts INTEGER n as parameter.
#

def viralAdvertising(n):
    if n>=1 and n<=50:
        cumulata=0
        a=5
        for i in range(n):
            cumulata+=a//2
            a=(a//2)*3
        return cumulata

    # Write your code here

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

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

    result = viralAdvertising(n)

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

    fptr.close()
```

## 79 Recursive Digit Sum

```
[ ]: #!/bin/python3

import math
import os
import random
```

```

import re
import sys

#
# Complete the 'superDigit' function below.
#
# The function is expected to return an INTEGER.
# The function accepts following parameters:
# 1. STRING n
# 2. INTEGER k
#

def superDigit(n, k):
    if k>=1 and k<=10**5:
        p=sum((int(i) for i in str(n)))*k
        if len(str(p))==1:
            return p
        else:
            k=1
            return superDigit(p, k)

    # Write your code here

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

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

    n = first_multiple_input[0]

    k = int(first_multiple_input[1])

    result = superDigit(n, k)

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

    fptr.close()

```

## 80 Insertion Sort - Part 1

```
[ ]: #!/bin/python3
```

```
import math
```

```

import os
import random
import re
import sys

#
# Complete the 'insertionSort1' function below.
#
# The function accepts following parameters:
# 1. INTEGER n
# 2. INTEGER_ARRAY arr
#

def insertionSort1(n, arr):
    if n>=1 and n<=1000:
        for i in range(n-1,0,-1):
            if arr[i]>=-10000 and arr[i]<=10000:
                if arr[i]<arr[i-1]:
                    a=arr[i]
                    arr[i]=arr[i-1]
                    print(*arr)
                    arr[i-1]=a

        print(*arr)

if __name__ == '__main__':
    n = int(input().strip())

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

    insertionSort1(n, arr)

```

## 81 Insertion Sort - Part 2

```

[ ]: #!/bin/python3

import math
import os
import random
import re
import sys

#
# Complete the 'insertionSort2' function below.
#
# The function accepts following parameters:
# 1. INTEGER n

```

```

# 2. INTEGER_ARRAY arr
#

def insertionSort2(n, arr):
    k=0
    while k<n-1:
        j=k+1
        a=arr[j]
        for i in range(j):
            if arr[i]>arr[j]:
                for o in range(j,i,-1):
                    arr[o]=arr[o-1]
                arr[i]=a
        print(*arr)
        k+=1

if __name__ == '__main__':
    n = int(input().strip())

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

    insertionSort2(n, arr)

```

## 82 Detect Floating Point Number

```

[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT

T=int(input())
if T>0 and T<10:
    for i in range (T):
        try:
            print(bool(float(input())))
        except:
            print(False)

```

## 83 Re.split()

```

[ ]: regex_pattern = r"[.,]" # Do not delete 'r'.

```

## 84 Group(), Groups() & Groupdict()



```
[ ]: import re
S=input()
match=re.search(r"([a-zA-Z0-9])\1+", S)
print(match.group()[0] if match else -1)
```

## 85 Re.findall() & Re.finditer()

```
[ ]: import re
S=input()
pattern=re.finditer(r'(?
↳<=[QWRTYPSDFGHJKLZXCVBNMqwrtypsdfghjklzxcvbnm])([AEIOUaeiou]{2,})?(?
↳<=[QWRTYPSDFGHJKLZXCVBNMqwrtypsdfghjklzxcvbnm])', S)
match=[i for i in map(lambda x: x.group(), pattern)]
print(*match, sep='\n') if match != [] else print(-1)
```

## 86 Re.start() & Re.end()

```
[ ]: # Enter your code here. Read input from STDIN. Print output to STDOUT
import re
S=input()
k=input()
l=l = len(k)-1
if len(S)>0 and len(S)<100 and len(k)>0 and len(k)<100:
    i=re.finditer(f'(?={k})', S)
    lista=[(a.start(), a.start()+1) for a in i]
    print(*lista or [(-1, -1)], sep = '\n')
```

## 87 Validating phone numbers

```
[ ]: import re
N=int(input())

if N>=1 and N<=10:
    for i in range(N):
        if re.match(r'^[789]\d{9}$', input()):
            print('YES')
        else:
            print('NO')
```

## 88 Hex Color Code

```
[ ]: import re
N=int(input())
a=re.compile(r'(?![^])#[0-9A-Fa-f]{3,6}')
if N>0 and N<50:
    for i in range (N):
        m = a.findall(input())
        if m:
            print(*m, sep='\n')
```

## 89 Validating and Parsing Email Addresses

```
[ ]: import email.utils, re
n=int(input())

a= r"^([A-Za-z].+[\@]{1}([A-Za-z])+[\.]{1}[a-z]{1,3})$"
if n>0 and n<100:
    for i in range(n):
        s=input()
        r = re.search(a, email.utils.parseaddr(s)[1])
        if r:
            print(s)
```

## 90 Regex Substitution

```
[ ]: import re
N=int(input())

if N>0 and N<100:
    for i in range (N):
        s = re.sub(r'(?<=\s)&&(=?=\s)', "and", str(input()))
        print(re.sub(r'(?<=\s)\\|\\| (=?=\s)', "or", s))
```

## 91 Validating UID

```
[ ]: import re
T=int(input())
a = r"^(?=(?:.*[A-Z]){2})(?=(?:.*[0-9]){3})(?:([a-zA-Z0-9])(?!.*\1)){10}$"
for i in range(T):
    print("Valid" if re.match(a, input()) else "Invalid")
```

## 92 HTML Parser - Part 1

```
[ ]: from html.parser import HTMLParser

n = int(input())
htmls = [input() for i in range(int(n))]

class MyHTMLParser(HTMLParser):
    def handle_starttag(self, tag, attrs):
        print("Start :", tag)
        for attr in attrs:
            print(f"-> {attr[0]} > {attr[1]}")
    def handle_endtag(self, tag):
        print("End    :", tag)
    def handle_startendtag(self, tag, attrs):
        print("Empty :", tag)
        for attr in attrs:
            print(f"-> {attr[0]} > {attr[1]}")

a = MyHTMLParser()
a.feed("".join(htmls))
```

## 93 HTML Parser - Part 2

```
[ ]: from html.parser import HTMLParser

class MyHTMLParser(HTMLParser):
    def handle_comment(self, data):
        if '\n' in data:
            print(">>> Multi-line Comment")
            print(data)
        else:
            print(">>> Single-line Comment")
            print(data)

    def handle_data(self, data):
        if data != '\n':
            print(">>> Data")
            print(data)

html = ""
for i in range(int(input())):
    html += input().rstrip()
    html += '\n'

parser = MyHTMLParser()
parser.feed(html)
```

```
parser.close()
```

## 94 Detect HTML Tags, Attributes and Attribute Values

```
[ ]: import re

html = ''.join(input() for _ in range(int(input())))
html = re.sub(r'<!--.*?-->', '', html)

tag = r'\s*<(\w+)'
attribute = r'([^\s]+)="(.*)"'
```

```
for tag, attr, value in re.findall(f'{tag}|{attribute}', html):
    print(tag or f'-> {attr} > {value}')
```