Untitled

Umberto Junior Mele

06 novembre 2016

Introduction

```
print("Hello, World!")
```

Reading Raw Input

```
s=input()
if len(s)>=1 and len(s)<=500:
    print(s)
```

Python If-Else

```
import sys
N = int(input().strip())
if N%2==1:
    print('Weird')
elif N>=2 and N<=5:
    print('Not Weird')
elif N>=6 and N<=20:
    print('Weird')
elif N>=20:
    print('Not Weird')
```

Arithmetic Operators

```
i=int(input())
l=int(input())
print(i+l)
print(i-l)
print(i*l)
```

Python: Division

```
i=float(input())
l=float(input())
print(int(i/l))
print(i/l)
```

Loops

```
l=int(input())
for i in range(l):
    print(i**2)
```

Write a function

```
def is_leap(year):
    leap = False

    # Write your logic here
    if int(year)%4==0 and (int(year)%100!=0 or int(year)%400==0):
        leap= True
    return leap
```

Print Function

```
print(*range(1, int(input())+1), sep='')
```

Basic Data Types

List

```
num=int(input())
list=[]
for _ in range(num):
    c = input().split()
    com = c[0]
    arg = c[1:]
    if com == 'print':
        print(list)
    else:
        com += '(' + ','.join(arg)+')'
        eval('list.'+com)
```

Tuples

```
l=input()
dati=input()
dati=dati.split()
dati= map(int,dati)
t=tuple(dati)
print(hash(t))
```

List Comprehension

```
x,y,z,n = (int(input()) for i in range(4))
print([[a,b,c]for a in range(x+1) for b in range(y+1) for c in range(z+1) if a+b+c != n])
```

Find the second larger number

```
s=int(input())
d=input()
lst=d.split(' ')
lst1=map(int,lst)
lst1 = list(lst1)
lst1.sort()
l=True
m=2
while l:
    if lst1[-m]==lst1[-1]:
        m+=1
    else:
        print (lst1[-m])
        break
```

Nested List

```
n=int(input())
lista=[[input(),float(input())]for _ in range(n)]
s=set([m for n,m in lista])
sls=sorted(list(s))[1]
print('\n'.join(sorted([nom for nom,v in lista if v==sls])))
```

Finding the Percentage

```
n=int(input())
diz={}
for _ in range(n):
    d=input().split()
    diz[d[0]]=d[1:]
last=input()
val=sum(map(float,diz[last]))/3.
print( "%.2f" % val)
```

Strings

sWAP cASE

```
n=input()
for i in range(len(n)):
    if ord(n[i])>= ord('A') and ord(n[i])<= ord('Z'):
        l= chr(ord('a')+(ord(n[i])-ord('A')))
```

```
    elif ord(n[i])>= ord('a') and ord(n[i])<= ord('z'):
        l= chr(ord('A')+(ord(n[i])-ord('a')))
    else:
        l=n[i]
    print(l, end='')
```

String Split and Join

```
n=input().split()
print('-'.join(n))
```

What's your Name?

```
print('Hello {} {}! You just delved into python.'.format(input(),input()))
```

Mutation

```
s=input()
l=input().split()
print(s[:int(l[0])]+l[1]+s[int(l[0])+1:])
```

Find a string

```
n,m=(input() for _ in range(2))
count=0
for i in range(len(n)-len(m)+1):
    p=True
    for g in range(len(m)):
        if n[i+g]!=m[g]:
            p=False
            break
    if p:
        count+=1
print(count)
```

String Validators

```
n=input()
print(any([l.isalnum() for l in n]))
print(any([l.isalpha() for l in n]))
print(any([l.isdigit() for l in n]))
print(any([l.islower() for l in n]))
print(any([l.isupper() for l in n]))
```

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
```

Text Wrap

```
import textwrap
print(textwrap.fill(input(),int(input())))
```

Designer Door Mat

```
N, M = map(int,input().split()) # More than 6 lines of code will result in 0 score. Blank lines are not
for i in range(1,N,2):
    print(('.|.'*i).center (M,'-')) #Enter Code Here
print('WELCOME'.center(M,'-')) #Enter Code Here
for i in range(N-2,-1,-2):
    print(('.|.'*i).center(M,'-')) #Enter Code Here
```

String Formatting

```
n=int(input())
w=len(bin(n)[2:])

for i in range(1,n+1):
    print('{0:{width}d} {0:{width}o} {0:{width}X} {0:{width}b}'.format(i,width=w))
```

Alphabet Rangoli

```
n=int(input())
L=[]
r=[]
for i in range(n):
    L.append(chr(ord('a')+i))
for j in range(n,-1,-1):
    r.append(L[n::-1][:n-j]+L[j+1:])
for k in [i for i in range(1,n)]+[h for h in range(n,0,-1)]:
        print(('-'.join(r[k])).center((n-1)*4+1,'-'))
```

Capitalize!!!

```
for i in input().split(' '): print(i.capitalize(), end=' ')
```

The Minion Game

```
n=input()
vow='AEIOU'
st=0
kev=0
for w in range(len(n)):
    if n[w] in vow:
        kev+=len(n)-w
    else:
        st+=len(n)-w
if kev>st: print('Kevin',kev)
elif st>kev: print('Stuart',st)
else: print('Draw')
```

Merge the tools

```
s,n=(input(),int(input()))
s=iter(s)
for i in zip(*[s]*n):
    d=dict()
    print(''.join([d.setdefault(c,c) for c in i if c not in d]))
```

Sets

Introduction to Sets

```
n,m=(input(),set(map(float,(input().split()))));print(sum(m)/len(m))
```

Symmetric Difference

```
_,n,_,m=(set(map(int,(input().split()))) for i in range(4))
dif= sorted(list(n.difference(m))+list(m.difference(n)))
for i in dif:print(i)
```

No Idea!

```
_,ar=(list(map(int, (input().split()))) for j in range(2))
a,b=(set(list(map(int, (input().split())))) for j in range(2))
hap=0
for el in ar:
    if el in a:
        hap+=1
    else: pass
    if el in b:
        hap-=1
    else: pass
print(hap)
```

set.add()

```
n=int(input())
S=list(str(input())for k in range(n))
print(len(set(S)))
```

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

```
n = int(input())
s = set(map(int, input().split()))
N= int(input())
Cm_l=list(input().split() for _ in range(N))
for o in range(N):
    if len(Cm_l[o])==1:
        eval("s."+Cm_l[o][0]+"()")
    else:
        eval("s."+Cm_l[o][0]+"("+str(Cm_l[o][1])+")")
print(sum(s))
```

Set .union() Operation

```
_,eng,_,fr=(set(input().split()) for _ in range(4))
print(len(eng|fr))
```

Set .intersection() Operation

```
(_,eng),(_,fr)=((input(),set(input().split())) for _ in range(2))
print(len(eng&fr))
```

Set .difference() Operation

```
(_,eng),(_,fr)=((input(),set(input().split())) for _ in range(2)) ; print(len(eng - fr))
```

Set .symmetric_difference() Operation

```
(_,eng),(_,fr)=((input(),set(input().split())) for _ in range(2)); print(len(eng^fr))
```

Set Mutations

```
_,A,n=(input(),set(input().split()), int(input()))
for _ in range(n):
    eval('A.'+input().split()[0]+'('+str(input().split())+')')
print(sum(map(int, A)))
```

The Captain's Room

```
n,A=(int(input()),list(map(int, input().split()))); k=set(A); print(int(((sum(k)*n) - (sum(A)))/(n-1)))
```

Check Subset

```
for i in range(int(input())): #More than 4 lines will result in 0 score. Blank lines won't be counted.
    a = int(input()); A = set(input().split())
    b = int(input()); B = set(input().split())
    print(len(A - B)==0)
```

Check Strict Superset

```
A=set(input().split())
z=1
for _ in range(int(input())): z*=(A > set(input().split()))
print(z==1)
```

Collections

collections.Counter()

```
import collections as c
_=input()
shoes=c.Counter(list(map(int,input().split())))
money=0
for _ in range(int(input())):
    l=list(map(int, input().split()))
    if( (int(l[0]) in shoes.keys()) and (int(shoes[int(l[0])])>0)) :
        shoes[l[0]]-=1
        money+= int(l[1])
print(money)
```

DefaultDict Tutorial

```
from collections import defaultdict
d = defaultdict(list)
n,s=(input().split())
l=[]
for i in range(1,int(n)+1):
    d[input()].append(i)
for _ in range(int(s)):
    z=input()
    if bool(d[z]):
        print(*d[z])
    else:
        print(-1)
```

Errors and Exceptions

Exceptions

```
for _ in range(int(input())):
    try:
        a, b = map(int, input().split())
        print(a//b)
    except (ZeroDivisionError, ValueError) as e:
        print("Error Code:", e)
```

Incorrect Regex

```
import re
for _ in range(int(input())):
    try:
        print(bool(re.compile(input())))
    except:
        print('False')
```

Regex and Parsing

Introduction to Regex Module

```
import re
print(*(bool(re.search('^[\+\-]?\d{0,}\.\d{1,}$', n)) for n in (input() for _ in range(int(input())))),
```

re.split()

```
import re
[print(i) for i in re.split('[.,]', input()) if i]
```

Group(), Groups() & Groupdict()

```
import re
m= re.search(r'([A-Za-z0-9])\1+',input().strip())
print(m.group(1) if m else -1)
```

Re.findall() & Re.finditer()

```
import re
n=re.findall(r'(?<=[^aeiou])([aeiou]{2,})(?=[^aeiou])',input(),re.I)
print('\n'.join(n  or ['-1']))
```

Re.start() & Re.end()

```
import re
n,s=(input() for _ in range(2))
[print('(%s, %s)' % (m.start(), m.end())) for m in re.finditer(s[:-1] + '(?=' + s[-1] + ')', n)] or pri
```

Regex Substitution

```
import re
def fc(match):
    if match.group()=='&&': return 'and'
    elif match.group()=='||': return r'or'

print(*[re.sub(r'(?<= )(&&|\|\|)(?= )',fc ,input()) for i in range(int(input()))],sep='\n')
```

Validation Roman Numerals

```
import re
print(bool(re.match('M{0,3}(CM|CD|D?C{0,3})(XC|XL|L?X{0,3})(IX|IV|V?I{0,3})$',input())))
```

Validation phone numbers

```
import re
print(*[ "YES" if (re.match(r"^[789]\d{9}$", input())) else "NO" for i in range(int(input())) ], sep='\
```

Validation and Parsing Email Addresses

```
import re
import email.utils as eu
for _ in range(int(input())):
    n=eu.parseaddr(input())
    if bool(re.match('^(?P<name>[A-Za-z][\d\w\-\.]+)@([A-Za-z]+)\.([a-z]{1,3})$',n[1])):
        print(eu.formataddr((n[0],n[1])))
```

Hex Color Code

```
import re, sys
[print(j) for i in sys.stdin for j in re.findall('[\s:](#[a-f0-9]{6}|#[a-f0-9]{3})', i, re.I)]
```

HTML Parser - Part 1

```
import re

quote = re.compile(r'<(!--.*?-->|<([^>]+)>', re.MULTILINE)
attribute = re.compile(r'^(\w+)\b(.*)$', re.MULTILINE)
value = re.compile(r'\s([^=\s]+)(?:=[\'"]?([^\'"]+)[\'"]?)?', re.MULTILINE)
text = ''

n = int(input())
for _ in range(n):
    s = input()
    text += s

m = quote.findall(text)

for u in m:
    if len(u[1]) > 0:
        t = u[1]
        if t.startswith('/'):
            print('End   : {0:s}'.format(t[1:]))
        else:
            q = attribute.match(t)
            if t.endswith('/'):
                print('Empty : {0:s}'.format(q.group(1).strip()))
            else:
                print('Start : {0:s}'.format(q.group(1).strip()))
            if len(q.group(2)) > 1:
                r = value.findall(q.group(2))
                for e in r:
                    if e[0] != '/':
                        print('-> {0:s} > {1:s}'.format(e[0], e[1] if len(e[1]) > 0 else 'None'))
```

Numpy

Arrays

```
import numpy as np
print(np.array(input().split(), float)[::-1])
```

Shape and Reshape

```
import numpy
print(numpy.reshape(numpy.array(input().split(),int),(3,3)))
```

Transpose and Flatten

```
import numpy as np
r,_=map(int,input().split())
A=np.array([input().split() for _ in range(r)],int)
print(np.transpose(A), A.flatten(), sep='\n')
```

Concatenate

```
import numpy as np
N,M,_=map(int,input().split())
A=np.array([input().split() for _ in range(N)],int)
B=np.array([input().split() for _ in range(M)],int)
print(np.concatenate((A,B),axis=0))
```

Zeros and Ones

```
import numpy
n=list(map(int,input().split()))
print(numpy.zeros(n,dtype=numpy.int),numpy.ones(n,dtype=numpy.int),sep='\n')
```

Eye and Identy

```
import numpy
r,s=map(int,input().split())
print(numpy.eye(r,s,k=0))
```

Array Mathemathics

```
import numpy
r,_=map(int,input().split())
A=numpy.array([input().split() for _ in range(r)],int)
B=numpy.array([input().split() for _ in range(r)],int)
print(A+B, A-B,A*B,A//B,A%B,A**B,sep='\n')
```

Floor, Ceil and Floor

```
import numpy as np
r=np.array(input().split(),float)
print(np.floor(r),np.ceil(r),np.rint(r), sep='\n')
```

Sum and Prod

```
import numpy as np
r,_=map(int, input().split())
A=np.array([input().split() for _ in range(r)], int)
print(np.prod(np.sum(A, axis=0)))
```

Min and Max

```
import numpy as np
r,_=map(int, input().split())
A= np.array([input().split() for _ in range(r)], int)
print(np.max(np.min(A,axis=1)))
```

Mean, var and std

```
import numpy as np
r,_=map(int,input().split())
A=np.array([input().split() for _ in range(r)], int)
print(np.mean(A, axis=1),np.var(A,axis=0),np.std(A),sep='\n')
```

Shape and reshape

```
import numpy
print(numpy.reshape(numpy.array(input().split(),int),(3,3)))
```

Dot and Cross

```
import numpy as np
r= int(input())
A=np.array([input().split() for _ in range(r)],int)
B=np.array([input().split() for _ in range(r)],int)
print(np.dot(A,B))
```