04-12-2020

**Say "Hello, World!" With Python**

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

**print**("Hello, World!")

# Python If-Else

*#!/bin/python*

**import** math

**import** os

**import** random

**import** re

**import** sys

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")

**elif** n%2 == 0 **and** n > 20:

**print**("Not Weird")

05-12-2020

**Arithmetic Operators**

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

    a = **int**(**raw\_input**())

    b = **int**(**raw\_input**())

**print**(a+b)

**print**(a-b)

**print**(a\*b)

**Python: Division**

**from** \_\_future\_\_ **import** division

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

    a = **int**(**raw\_input**())

    b = **int**(**raw\_input**())

**print**(a//b)

**print**(a/b)

**Loops**

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

    n = **int**(**input**())

    i = 0

**while** n > 0:

**print**(i \* i)

        i += 1

        n -= 1

**Print Function**

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

    n = **int**(**input**())

**for** i **in** **range**(1,n+1):

**print**(**str**(i),end = "")

**Write a function**

**def** is\_leap(year):

    leap = **False**

**if** year % 400 == 0 :

        leap = **True**

**elif** year % 100 == 0 :

        leap = **False**

**elif** year % 4 == 0:

        leap = **True**

**else**:

        leap = **False**

*# Write your logic here*

**return** leap

year = **int**(**input**())

06-12-2020

**List Comprehensions**

x = **int**(**input**())

y = **int**(**input**())

z = **int**(**input**())

n = **int**(**input**())

**print**([[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])

09-12-2020

**Lists**

N = **int**(**input**())

**list** = []

**for** i **in** **range**(N):

    condition = **str**(**input**())

**if** "insert" **in** condition:

        position = **int**(condition[7])

        value = **int**(condition[9:])

**list**.insert(position,value)

**if** "print" **in** condition:

**print**(**list**)

**if** "remove" **in** condition:

        position = **int**(condition[7:])

**list**.remove(position)

**if** "append" **in** condition:

        position = **int**(condition[7:])

**list**.append(position)

**if** "sort" **in** condition:

**list**.sort()

**if** "reverse" **in** condition:

**list**.reverse()

**if** "pop" **in** condition:

**list**.pop()

15-12-2020

**Find the Runner-Up Score!**

x = **int**(**input**())

arr = **set**(**list**(**map**(**int**,**input**().split())))

arr1 = **sorted**(arr)

**print**(arr1[-2])

**Nested Lists**

l = []

second\_lowest\_names = []

scores = **set**()

**for** \_ **in** **range**(**int**(**input**())):

    name = **input**()

    score = **float**(**input**())

    l.append([name, score])

    scores.add(score)

second\_lowest = **sorted**(scores)[1]

**for** name, score **in** l:

**if** score == second\_lowest:

        second\_lowest\_names.append(name)

**for** name **in** **sorted**(second\_lowest\_names):

**print**(name, end='\n')

**sWAP cASE**

**def** swap\_case(s):

**return** s.swapcase()

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

    s = **input**()

    result = swap\_case(s)

**print**(result)

16-12-2020

**String Split and Join**

**def** split\_and\_join(line):

    line = line.split()

**return** "-".join(line)

*# write your code here*

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

    line = **input**()

    result = split\_and\_join(line)

**print**(result)

**Tuples**

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

    n = **int**(**input**())

    integer\_list = **map**(**int**, **input**().split())

**print**(**hash**(**tuple**(integer\_list)))

**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**()

**for** keys **in** student\_marks:

**if** keys == query\_name:

            result = **sum**(student\_marks[keys])/**len**(student\_marks[keys])

**print**("{:.2f}".**format**(result))

**What's Your Name?**

**def** print\_full\_name(a, b):

**print**("Hello " +first\_name+" "+last\_name+"! You just delved into python.")

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

    first\_name = **input**()

    last\_name = **input**()

    print\_full\_name(first\_name, last\_name)

**Mutations**

**def** mutate\_string(string, position, character):

**return** string[:position] + character + string[position+1:]

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

    s = **input**()

    i, c = **input**().split()

    s\_new = mutate\_string(s, **int**(i), c)

**print**(s\_new)

**Find a string**

**def** count\_substring(string, sub\_string):

    count = 0

    n = **len**(string)+**len**(sub\_string)

**for** i **in** **range**(n):

**if** (string[i:**len**(sub\_string)+i]) == sub\_string:

            count += 1

**return** count

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

    string = **input**().strip()

    sub\_string = **input**().strip()

    count = count\_substring(string, sub\_string)

**print**(count)

23-12-2020

**Collections.OrderedDict()**

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

**from** collections **import** OrderedDict;

N = **int**(**input**())

d = OrderedDict()

**for** i **in** **range**(N):

    item = **input**().split()

    itemPrice = **int**(item[-1])

    itemName = " ".join(item[:-1])

**if**(d.get(itemName)):

        d[itemName] += itemPrice

**else**:

        d[itemName] = itemPrice

**for** i **in** d.keys():

**print**(i, d[i])

01-01-2021

# Collections : word order

n = **int**(**input**())

list1 = []

list2 = []

**for** i **in** **range**(n):

    s = **input**()

    list1.append(s)

**set** = **set**(list1)

list3 = **list**(**set**)

**for** i **in** list3:

    count = 0

**for** j **in** list1:

**if** i == j:

            count += 1

    list2.append(count)

**print**(**len**(list2))

list2.sort(reverse = **True**)

**for** i **in** **range**(**len**(list2)):

**print**(list2[i],end = " ")

time limit exceeded

07-01-2021

**Text Wrap**

**import** textwrap

**def** wrap(string, max\_width):

**return** textwrap.fill(string, width = max\_width)

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

    string, max\_width = **input**(), **int**(**input**())

    result = wrap(string, max\_width)

**print**(result)

08-01-2021

**Capitalize!**

*#!/bin/python3*

**import** math

**import** os

**import** random

**import** re

**import** sys

*# Complete the solve function below.*

**def** solve(s):

    s = ' '.join(**map**(**str**.capitalize, s.split(' ')))

**return** s

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

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

    s = **input**()

    result = solve(s)

    fptr.write(result + '\n')

    fptr.close()

09-01-2021

# Numpy : Arrays

**import** numpy

**def** arrays(arr):

    a = numpy.array(arr,**float**)

    result = a[::-1]

**return** result

*# complete this function*

*# use numpy.array*

arr = **input**().strip().split(' ')

result = arrays(arr)

**print**(result)

13-01-2021

**Text Alignment**

*#Replace all \_\_\_\_\_\_ with rjust, ljust or center.*

thickness = **int**(**raw\_input**()) *#This must be an odd number*

c = 'H'

*#Top Cone*

**for** i **in** **range**(thickness):

**print** (c\*i).\_\_\_\_\_\_(thickness-1)+c+(c\*i).\_\_\_\_\_\_(thickness-1)

*#Top Pillars*

**for** i **in** **range**(thickness+1):

**print** (c\*thickness).\_\_\_\_\_\_(thickness\*2)+(c\*thickness).\_\_\_\_\_\_(thickness\*6)

*#Middle Belt*

**for** i **in** **range**((thickness+1)/2):

**print** (c\*thickness\*5).\_\_\_\_\_\_(thickness\*6)

*#Bottom Pillars*

**for** i **in** **range**(thickness+1):

**print** (c\*thickness).\_\_\_\_\_\_(thickness\*2)+(c\*thickness).\_\_\_\_\_\_(thickness\*6)

*#Bottom Cone*

**for** i **in** **range**(thickness):

**print** ((c\*(thickness-i-1)).\_\_\_\_\_\_(thickness)+c+(c\*(thickness-i-1)).\_\_\_\_\_\_(thickness)).\_\_\_\_\_\_(thickness\*6)

15-01-2021

**itertools.product()**

**from** itertools **import** product

l1 = **list**(**map**(**int**,**input**().split()))

l2 = **list**(**map**(**int**,**input**().split()))

a = [l1,l2]

**print**(**list**(product(\*a)))

18-01-2021

**collections.Counter()**

**import** collections

X = **int**(**input**())

shoes = collections.Counter(**map**(**int**,**input**().split()))

income = 0

N  = **int**(**input**())

**for** i **in** **range**(N):

    size,price = **input**().split()

**if** shoes[**int**(size)]:

        income += **int**(price)

        shoes[**int**(size)] -= 1

**print**(income)

19-01-2021

**Input()**

x,y = **map**(**int**,**input**().split())

**print**(**eval**(**input**())==y)

22-01-2021

**Detect Floating Point Number**

**import** re

**for** i **in** **range**(**int**(**input**())):

    N = **input**()

**print**(**bool**(re.match('^[-+]?[0-9]\*\.[0-9]+$',N)))

<https://realpython.com/regex-python/>

24-01-2021

**itertools.permutations()**

**from** itertools **import** permutations

s,n = **input**().split()

N = **int**(n)

a = **list**(permutations(s,N))

a.sort()

**for** i **in** a:

**print**(''.join(i))

**itertools.combinations()**

**from** itertools **import** combinations

s,n = **input**().split()

N = **int**(n)

**for** N **in** **range**(1,N+1):

    s1 = **list**(s)

    s1.sort()

    a = **list**(combinations(s1,N))

    a.sort()

**for** i **in** a:

**print**(''.join(i))