

15th June 2019

Hacker Earth Problem

**** Explanation ****

- li = [3,2,-1,-2,-3] (Original List)
- Sort the data
- li = [-3,2,-1,2,3] (Sorted List)
- pl = [1,2,2,3,3] (Positive Sorted List)
- p1[0] ----> Check if this number is -ve or +ve in the Original List
- if p1[0] in li:
 - return pl[0]
- else
 - return -p1[0]

Problem

```
In [23]: # Hacker Earth problem solvation
li = [-1, -2, 2, 3, 1]

li.sort()

p = []

for i in li:
    p.append(abs(i))
p.sort()
if p[0] in li:
    print(p[0]) #print(-p[0]) if less value given
else:
    print(-p[0]) # print(p[0]) if less value print means
```

-1

```
In [27]: # Farthest From Zero
li=[-1,-2,2,3,1,-100]

li.sort()

p = []

for i in li:
    p.append(abs(i))
p.sort()
if p[-1] in li:
    print(p[-1]) #print(-p[0]) if less value given
else:
    print(-p[-1]) # print(p[0]) if less value print means
```

-100

Problem 3

```
In [48]: # you are given three numbers,a,b,c.Write a program to find the Largest number
# and Leaves remainder b when divided by a if not print -1
```

```
def greater(a,b,c):
    for i in range(c,b-1,-1):
        if (i<=c) and (i%a==b):
            return i
    else:
        return -1
greater(3,2,9)
```

Out[48]: 8

```
In [47]: # you are given three numbers,a,b,c.Write a program to find the Largest number
# and Leaves remainder b when divided by a if not print -1
```

```
def greater(a,b,c):
    for i in range(c,b+1,-1): # c,c-1,c-2.....a
        if (i<=c) and (i%a==b):
            return i
    else:
        return -1
greater(1,2,4)
```

Out[47]: -1

In [50]: *# you are given three numbers,a,b,c.Write a program to find the Largest number
and Leaves remainder b when divided by a if not print -1*

```
def greater(a,b,c):  
    for i in range(c,b-1,-1):  
        if (i%a==b):  
            return i  
    else:  
        return -1  
greater(3,2,100)
```

Out[50]: 98

In []:

In []:

```
In [1]: dir(list())
```

```
Out[1]: ['__add__',
         '__class__',
         '__contains__',
         '__delattr__',
         '__delitem__',
         '__dir__',
         '__doc__',
         '__eq__',
         '__format__',
         '__ge__',
         '__getattr__',
         '__getitem__',
         '__gt__',
         '__hash__',
         '__iadd__',
         '__imul__',
         '__init__',
         '__init_subclass__',
         '__iter__',
         '__le__',
         '__len__',
         '__lt__',
         '__mul__',
         '__ne__',
         '__new__',
         '__reduce__',
         '__reduce_ex__',
         '__repr__',
         '__reversed__',
         '__rmul__',
         '__setattr__',
         '__setitem__',
         '__sizeof__',
         '__str__',
         '__subclasshook__',
         'append',
         'clear',
         'copy',
         'count',
         'extend',
         'index',
         'insert',
         'pop',
         'remove',
         'reverse',
         'sort']
```

```
In [13]: l=[1,2,3,4,5]
l.append(22)
l
l.append([9,8,7])
l
l.extend([1,2])
l
l.insert(0,[55,33,44])
l
```

```
Out[13]: [[55, 33, 44], 1, 2, 3, 4, 5, 22, [9, 8, 7], 1, 2]
```

```
In [ ]: a=int(input())
b=int(input())
c=int(input())
i=c
while i!=0:
    if i<=c and i%a==b:
        print(i)

        # break
    else:
        print(-1)
        #break
        i=i-1
```

```
In [5]: # Function to generate the prime and fibanocci series and print the data in fir

n=int(input())
plist = []
for i in range (n):
    plist.append(input())
print(plist)
#primefibdata(6)
```

```
2
sya
mkf
['sya', 'mkf']
```

```
In [ ]: li=[int(i) for i in input().split()][:4]
li
```

In [17]: *# function to count the count divisor in hacker earth problems*

```
def countdivisor(l,r,k):
    count=0
    for i in range(l,r+1):
        if(i%k==0):
            count=count+1
    print (count)
```

```
li=input().split()
l=int(li[0])
r=int(li[1])
k=int(li[2])
countdivisor(l,r,k)
```

```
1 10 1
10
```

In [22]: **def** countdivisor(l,r,k):

```
    c=0
    for n in range (l,r+1):
        if n%k==0:
            c=c+1
    return c
```

```
s=input()
s=s.split()
```

```
l=int(s[0])
r=int(s[1])
k=int(s[2])
```

```
countdivisor(l,r,k)
```

```
1 10 1
```

Out[22]: 10

In [24]: *# function to print the factorial of a given number in a hacker earth problem*

```
def factorial(n):
    fact=1
    for i in range(1,n+1):
        fact=fact*i
    print(fact)
n=int(input())
factorial(n)
```

```
3
6
```

In [26]: *# Function to print the given string is palindrome YES or not if print NO*

```
def palindrome(string):
    if string == string[::-1]:
        print("YES")
    else:
        print("NO")
string=input()
palindrome(string)
```

aba
YES

In [27]: `ord('a')`

Out[27]: 97

In [28]: `chr(97)`

Out[28]: 'a'

In [32]: `ord('d')-96`

Out[32]: 100

```
In [2]: def a(n):
        for i in n:
            if i==i.upper():
                a=i.lower()
            elif i==i.lower():
                a=i.upper()
            print(a,end=" ")

n=input()
a(n)
```

aBcDeF
AbCdEf

In [11]: *# Function to print the*

```
def string(s):
    string1=s.swapcase()
    print(string1)
s=input()
string(s)
```

AbCdEf
aBcDeF

```
In [20]: ord('Z')+32
```

```
Out[20]: 122
```

```
In [ ]:
```

```
In [14]: def prime_range(n):  
         for i in range(1,n+1):
```

```
5
```

```
In [1]: def Duration(sh,sm,eh,em):  
        a=(sh*60)+sm  
        b=(eh*60)+em  
        c=b-a  
        d=c//60  
        e=c%60  
        print(d,end=" ")  
        print(e)  
  
        n=int(input())  
        while(n):  
            s=input().split()  
            sh=int(s[0])  
            sm=int(s[1])  
            eh=int(s[2])  
            em=int(s[3])  
            Duration(sh,sm,eh,em)  
            n=n-1
```

```
2  
1 44 2 14  
0 30  
2 42 8 23  
5 41
```



```
In [6]: def Duration(sh,sm,eh,em):
        a=(sh*60)+sm
        b=(eh*60)+em
        c=b-a
        d=c//60
        e=c%60
        print(d, end=" ")
        print(e, end="\n")
        return

        n=int(input())
        for i in range(n):
            s=input().split()
            sh=int(s[0])
            sm=int(s[1])
            eh=int(s[2])
            em=int(s[3])

            Duration(sh,sm,eh,em)
```

```
2
1 44 2 14
0 30
2 42 8 23
5 41
```

```
In [22]: name = input()                # Reading input from STDI
        print('%s'% name)
```

```
eeeeee
eeeeee
```

```
In [2]: T=int(input())
def tw(s,t):
    f=1
    if(len(s)!=len(t)):
        return 'NO'
    else:
        for i in range(len(s)):
            if (s.count(s[i])!=t.count(s[i])):
                return 'NO'
        if f==1:
            return 'YES'
for i in range (T):
    st=input().split()
    s=st[0]
    t=st[1]
    print(tw(s,t))
```

```
1
syamala aysmala
YES
```

```
In [ ]:
```

```
In [ ]:
```

```
In [9]: n=int(input())
def tw(s1,s2):
    if (len(s1)!=len(s2)):
        return "NO"
    elif sorted(s1)==sorted(s2):
        return "YES"
    else:
        return "NO"
for i in range(n):
    st=input().split()
    s1=st[0]
    s2=st[1]
    rs=tw(s1,s2)
    print(rs)
```

```
1
vdf grgrg
NO
```

```
In [ ]:
```