Contribution to geeks for geeks : https://ide.geeksforgeeks.org/crv5srFJaQ

Python problems

1. class Employee(object):

def \_\_init\_\_(self,employee\_name):

self.employee\_name=employee\_name

def calculate\_wage(self,hours):

self.hours=hours

return hours\*20.00

class PartTimeEmployee(Employee):

def calculate\_wage(self,hours):

self.hours=hours

return hours\*12.00

def full\_time\_wage(self,hours):

return super(PartTimeEmployee,self).calculate\_wage(hours)

milton=PartTimeEmployee("milton")

print(milton.full\_time\_wage(10))

o/p : 200.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. class Calculate(object):

def \_\_init\_\_(self,n1,n2):

self.n1=n1

self.n2=n2

def add(self):

return self.n1+self.n2

def sub(self):

return self.n1-self.n2

x=Calculate(5,6)

print(x)

<\_\_main\_\_.Calculate object at 0x7ff4452a1590>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

3. def substrCount(n, s):

c=0

for i in range(n):

x=""

y=""

for j in range(i,n):

x = x+s[j]

y=x[::-1]

if x==y:

c+=1

return c

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. class Employee(object):

def \_\_init\_\_(self,employee\_name):

self.employee\_name=employee\_name

def calculate\_wage(self,hours):

self.hours=hours

return hours\*20.00

class PartTimeEmployee(Employee):

def calculate\_wage(self,hours):

self.hours=hours

return hours\*12.00

def full\_time\_wage(self,hours):

return super().calculate\_wage(hours)

milton=PartTimeEmployee("milton")

print(milton.full\_time\_wage(10))

#print(milton.calculate\_wage(10))

200.0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.balanced brackets :

Imp : **"string":"((){{{{[]}}}})";**

def balancedBrackets(string):

stack = []

for i in string:

if i =='(' or i == '{' or i == '[':

stack.append(i)

else:

if stack == []:

return False

previous = stack[-1]

if i ==')' and previous == '(':

stack.pop()

elif i =='}' and previous == '{':

stack.pop()

elif i ==']' and previous == '[':

stack.pop()

else:

return False

return len(stack) == 0

return False

**"string":"((){{{{[]}}}})";**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.1

s="()(()()"

stack=[]

n=len(s)

count=0

for i in s:

print("stack at present : ",stack)

if i=="(":

stack.append(i)

continue

else:

if stack==[]:

continue

else:

if stack.pop()=="(" and i==")":

count+=1

print(count)

print(stack)

**$python3 main.py**  
stack at present : []

stack at present : ['(']

stack at present : []

stack at present : ['(']

stack at present : ['(', '(']

stack at present : ['(']

stack at present : ['(', '(']

3

['(']

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.2.

s="))())()))(()"

stack=[]

c=0

for i in s:

if i in "(":

stack.append(i)

print(stack)

else:

if stack==[]:

continue

else:

check=stack.pop()

print("checked"+check)

if i == ')':

c=c+2

print(c)

o/p :

['(']

checked(

['(']

checked(

['(']

['(', '(']

checked(

6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5.3

s="((()()()()(((())"

stack=[]

n=len(s)

count=0

for i in s:

print("stack at present : ",stack)

if i=="(":

stack.append(i)

continue

else:

if stack==[]:

continue

else:

if stack.pop()=="(" and i==")":

count+=1

print(count\*2)

print(stack)

**$python3 main.py**  
stack at present : []

stack at present : ['(']

stack at present : ['(', '(']

stack at present : ['(', '(', '(']

stack at present : ['(', '(']

stack at present : ['(', '(', '(']

stack at present : ['(', '(']

stack at present : ['(', '(', '(']

stack at present : ['(', '(']

stack at present : ['(', '(', '(']

stack at present : ['(', '(']

stack at present : ['(', '(', '(']

stack at present : ['(', '(', '(', '(']

stack at present : ['(', '(', '(', '(', '(']

stack at present : ['(', '(', '(', '(', '(', '(']

stack at present : ['(', '(', '(', '(', '(']

12

['(', '(', '(', '(']

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. nums=[0,1,2,2,3,0,4,2]

val=2

c=0

for i in range(len(nums)):

if val in nums:

nums.remove(val)

print(nums)

o/p:

[0, 1, 3, 0, 4]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

7.

s="pwwkew"

s1=""

max=""

for i in s:

if i not in s1:

s1=s1+i

print("string is : "+s1)

if len(s1)>len(max):

max=s1

else:

s1=""

s1=s1+i

if len(s1)>len(max):

max=s1

print(max)

o/p string is : p

string is : pw

string is : wk

string is : wke

wke

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

8. a=[0, -1, 2, -3, 1]

n=len(a)

c=0

i=0;

j=i+1

k=j+1

sum=a[i]+a[j]+a[k]

while i<n:

if k==n-1:

j=j+1

k=j+1

if j==n-1:

i=i+1

j=i+1

k=j+1

if sum!=0:

sum=sum-a[k]

print("sum after deletion is : "+str(sum))

print("value in list is " + str(a[k]))

k+=1

if k<n:

sum=sum+a[k]

print("sum here is "+str(sum) )

print(str(i)+" "+str(j)+" "+str(k))

if sum==0:

c+=1

i=i+1

j+=1

k=j+1

sum=a[i]+a[j]+a[k]

print(sum)

print("the pairs ARE : "+str(c))

o/p:

sum after deletion is : -1

value in list is 2

sum here is -4

0 1 3

sum after deletion is : -1

value in list is -3

sum here is 0

0 1 4

-2

sum after deletion is : 1

value in list is -3

sum here is 2

1 2 4

sum after deletion is : 1

value in list is 1

Traceback (most recent call last):

File "main.py", line 17, in <module>

sum=sum-a[k]

IndexError: list index out of range

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

9.

a = [-1,0,1,2,-1,-4]

n=len(a)

a.sort()

for i in range(0,n-2):

val1=a[i]

val2=-(val1)

sp=i+1

ep=n-1

sum=0

while(sp<ep):

sum=a[sp]+a[ep]

if sum==val2:

print("got it")

print(str(a[i])+" "+str(a[sp])+" "+str(a[ep]))

break

elif sum>val2:

ep-=1

elif sum<val2:

sp+=1

o/p :

got it

-1 -1 2

got it

-1 0 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

10. import string

s="The quick brown fox jumps over the dog"

x=list(s)

print(x)

a=[]

b={}

a = list(string.ascii\_lowercase)

b=a

print(a)

print(b)

o/p :

['T', 'h', 'e', ' ', 'q', 'u', 'i', 'c', 'k', ' ', 'b', 'r', 'o', 'w', 'n', ' ', 'f', 'o', 'x', ' ', 'j', 'u', 'm', 'p', 's', ' ', 'o', 'v', 'e', 'r', ' ', 't', 'h', 'e', ' ', 'd', 'o', 'g']

['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

11.

s="The quick brown fox jumps over the dog"

x=s.lower()

y=set(x)

if " " in y:

y.remove(" ")

n=len(y)

print(n)

print(y)

if n==26:

print("The String is a Pangram")

else:

print("The String is not a Pangram")

o/p :

22

{'f', 'w', 'k', 'o', 'p', 'd', 'j', 'v', 'n', 'x', 'e', 't', 's', 'i', 'h', 'g', 'c', 'r', 'q', 'm', 'u', 'b'}

The String is not a Pangram

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

12.

s="ABDEFGABEF"

max=0

n=len(s)

for i in range(n):

st=s[i]

for j in range(i+1,n):

if s[j] not in st:

st+=s[j]

else:

break

if len(st)>max:

max=len(st)

print(max)

o/p :

6

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

13.

import re

d={1:'one',2:'two',3:"three",4:'four',5:'five',6:'six',7:'seven',8:'eight',9:'nine'}

s="aeiou"

y=""

sum=0

x=[1,2,3,4,5]

n=len(x)

for i in x:

y=d[i]

print(y)

x1 = re.findall("[aeiou]", y)

print(x1)

sum+=len(x1)

print(sum)

i=0

j=n-1

target = sum

while(i<j):

if x[i]+x[j]>target:

j-=1

elif x[i]+x[j]<target:

i+=1

if x[i]+x[j]==target:

print("got it bro")

print(x[i],x[j])

break

o/p:

one

['o', 'e']

two

['o']

three

['e', 'e']

four

['o', 'u']

five

['i', 'e']

9

got it bro

4 5

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

14.

# 9 97 76 45 4 34 3 2

a=[2,34,3,97,9,76,45,4]

n=len(a)

for i in range(n):

if len(str(a[i]))==1:

a[i]=a[i]\*10+9

print(a)

x=[]

s=""

n1=len(a)-1

a.sort()

while(n1>=0):

num=str(a[n1])

size=len(num)-1

if num[size]=='9':

a[n1]=int(a[n1]/10)

s+=str(a[n1])

n1-=1

print(s)

o/p : [29, 34, 39, 97, 99, 76, 45, 49]

997764453342

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

15.

s = "bcabc"

s1=set(s)

print(s1)

o/p :

{'c', 'a', 'b'}

16.

s = "bcabc"

s1=set(s)

print(s1)

s2="".join(s1)

print(s2)

o/p:

{'b', 'a', 'c'}

bac

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IMP TCS 2021 march :

a=[20,1,6,8,13,20,1,7,20,13,16,20]

n=len(a)

x=20

n1=3

i=0

count=0

while(i<n):

print(i)

print()

for j in range(i,i+3):

print(a[j])

if(a[j]==20):

count+=1

print()

i=i+3

print(count)

0

20

1

6

3

8

13

20

6

1

7

20

9

13

16

20

4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

x.

s1="nice"

s2="chine"

n=len(s1)

s1=''.join(sorted(s1))

s2=''.join(sorted(s2))

i=0

c=0

while(i<n):

if s1[i]==s2[i] :

c+=1

i+=1

print(c==n)

False

17.

a=[1,2,3,4,5,6]

n=len(a)

t=10

for i in range(n):

for j in range(n):

if a[i]+a[j]==10:

print(a[i],a[j])

o/p:

4 6

5 5

6 4

18.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

a=[52,66,64,36,45,24,32]

n=len(a)

sum=0

for i in range(n):

x=a[i]

c=0

for j in range(i,n):

if a[j]>x:

c+=1

if c==0:

sum=sum+x

print(sum)

207

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

X.

s='aabbccd'

d={}

n=len(s)

for i in range(n):

if s[i] not in d:

d[s[i]]=1

else:

d[s[i]]+=1

print(d,end=' ')

{'a': 2, 'b': 2, 'c': 2, 'd': 1}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

X2.

s='aabbcccd'

d={}

n=len(s)

for i in range(n):

if s[i] not in d:

d[s[i]]=1

else:

d[s[i]]+=1

print(d,"\n")

max1=0

char=''

for i in range(n):

if d[s[i]]>max1:

max1=d[s[i]]

char=s[i]

print()

print(max1)

print(char)

**o/p:**

{'a': 2, 'b': 2, 'c': 3, 'd': 1}

3

c

x3.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

thisdict = {

"brand": "Ford",

"model": "Mustang",

"year": 1964

}

for x, y in thisdict.items():

print(x, y)

brand Ford  
model Mustang  
year 1964

s='aabbckkccd fhfhfhhh kjlkjjkkkkk'

d={}

n=len(s)

for i in range(n):

if s[i] not in d:

d[s[i]]=1

else:

d[s[i]]+=1

print(d,"\n")

max1=0

char=''

for i in range(n):

if d[s[i]]>max1:

max1=d[s[i]]

char=s[i]

print()

print(max1)

print(char)

for k,v in d.items():

print(k,v)

{'a': 2, 'b': 2, 'c': 3, 'k': 9, 'd': 1, ' ': 3, 'f': 3, 'h': 5, 'j': 3, 'l': 1}

9

k

a 2

b 2

c 3

k 9

d 1

3

f 3

h 5

j 3

l 1

a=[1,2,3,4,5,6,7,8,9]

n=len(a)

y=[]

i=0

while(i<n):

print(i)

x=a[i:i+3]

print(x)

x=x[::-1]

print(x)

y=y+x

i=i+3

print(y)

# [3,2,1,6,5,4,9,8,7]

0

[1, 2, 3]

[3, 2, 1]

3

[4, 5, 6]

[6, 5, 4]

6

[7, 8, 9]

[9, 8, 7]

[3, 2, 1, 6, 5, 4, 9, 8, 7]

a=[1,2,3,4,5,6,7,8,9]

n=len(a)

y=[]

i=0

while i<n:

x=a[i:i+3]

x=x[::-1]

y=y+x

i=i+3

print(y)

# [3,2,1,6,5,4,9,8,7]

a=[1,2,3,4,5,6,4,5,4]

n=len(a)

x=[]

for i in range(0,n,3):

print(i)

y=a[i:i+3]

print(y)

0

[1, 2, 3]

3

[4, 5, 6]

6

[4, 5, 4]

a=[1,2,3,4,5,6,4,5,4]

n=len(a)

x=[]

for i in range(0,n,3):

print(i)

y=a[i:i+3]

x.append(y)

print(y)

print('')

print(x)

print('')

for i in range(len(x)):

print(x[i])

if 4 in x[i]:

print("got it ")

0

[1, 2, 3]

3

[4, 5, 6]

6

[4, 5, 4]

[[1, 2, 3], [4, 5, 6], [4, 5, 4]]

[1, 2, 3]

[4, 5, 6]

got it

[4, 5, 4]

got it

----------------------------------------------------------------------------

a=[1,2,3,4,5,6,4,5,4]

n=len(a)/3

x=[]

for i in range(0,n):

print(i)

y=a[i:i+3]

x.append(y)

print(y)

print('')

print(x)

print('')

for i in range(len(x)):

print(x[i])

if 4 in x[i]:

print("got it ")

0

[1, 2, 3]

1

[2, 3, 4]

2

[3, 4, 5]

[[1, 2, 3], [2, 3, 4], [3, 4, 5]]

[1, 2, 3]

[2, 3, 4]

got it

[3, 4, 5]

got it

----------------------------------------------------------------------------

a="aabccbbccaaa"

d={}

s=''

for i in a:

if s=='' or i in s:

s=s+i

else:

x=len(s)

print(s[0],end='')

print(x,end='')

s=''

s=s+i

print(s[0],end='')

print(len(s),end='')

a2b1c2b2c2a3

subsequence.

import re

a="coronavirus"

x=3

b="coviruo"

n1=len(a)

n2=len(b)

i=0

j=0

while(i<n1 and j<n2):

if a[i]==b[j]:

j+=1

i+=1

print((j == n2))

False

import re

a="coronavirus"

x=3

b="coviru"

n1=len(a)

n2=len(b)

i=0

j=0

while(i<n1 and j<n2):

if a[i]==b[j]:

j+=1

i+=1

print((j == n2))

True

a=[13,45,21,65]

b=[13,21,45]

n1=len(a)

n2=len(b)

i=0

j=0

while(i<n1 and j<n2):

if b[j]==a[i]:

j+=1

i=i+1

print(j==n2)

False

''' Read input from STDIN. Print your output to STDOUT '''

#Use input() to read input from STDIN and use print to write your output to STDOUT

def main():

n=int(input())

while n>0:

a,b=input().split()

a=int(a)

b=int(b)

x=[]

for i in range(a,b+1):

if i>1:

for j in range(2,int(i/2)+1):

if i%j==0:

break

else:

x.append(i)

if len(x)==0:

print(-1)

elif len(x)==1:

print(0)

else:

print(x[len(x)-1]-x[0])

n-=1

main()

''' Read input from STDIN. Print your output to STDOUT '''

#Use input() to read input from STDIN and use print to write your output to STDOUT

def main():

n=int(input())

while n>0:

a,b=input().split()

a=int(a)

b=int(b)

x=[]

for i in range(a,b+1):

if i>1:

for j in range(2,int(i/2)+1):

if i%j==0:

break

else:

x.append(i)

if len(x)==0:

print(-1)

elif len(x)==1:

print(0)

else:

print(x[len(x)-1]-x[0])

n-=1

main()

|  |
| --- |
| # Python3 code to demonstrate  # backward iteration  # using reversed()    # Initializing number from which  # iteration begins  N = 6    # using reversed() to perform the back iteration  print ("The reversed numbers are : ", end = "")  for num in reversed(range(N + 1)) :      print (num, end = " ") |

**Output :**

The reversed numbers are : 6 5 4 3 2 1 0

---------------------------------------------------------------------

|  |
| --- |
| # Python3 code to demonstrate  # backward iteration  # using range(N, -1, -1)    # Initializing number from which  # iteration begins  N = 6  # using reversed() to perform the back iteration  print ("The reversed numbers are : ", end = "")  for num in range(N, -1, -1) :      print (num, end = " ") |

**Output :**

The reversed numbers are : 6 5 4 3 2 1 0

Tapping rain water :

a=[3,1,2,4,0,1,3,2]

n=len(a)

left=[]

right=[]

lmax=0

rmax=0

for i in range(n):

if a[i]>lmax:

lmax=a[i]

left.append(lmax)

print(left)

for j in range(n-1,-1,-1):

if a[j]>rmax:

rmax=a[j]

right.append(rmax)

right.sort(reverse=True)

print(right)

sum1=0

for i in range(n):

x=min(right[i],left[i])-a[i]

sum1+=x

print(sum1)

[3, 3, 3, 4, 4, 4, 4, 4]

[4, 4, 4, 4, 3, 3, 3, 2]

8

**--------------------------------------**

Implementing set without using inbuild functions :

nums = [6,6,6,6,7,7,7,7,9,9,9,10,10,10,55,55,55,88,88,88]

n=len(nums)

current=0

diff=0

i=0

y=[]

while i<n:

current=nums[i]

if nums[i]!=nums[i-1]:

diff=nums[i-1]

print(diff)

y.append(diff)

i+=1

y.sort()

print(y)

88

6

7

9

10

55

[6, 7, 9, 10, 55, 88]

----------------------------------------------------------------------------------------------------

To sort an set use

class Solution:

def removeDuplicates(self, nums: List[int]) -> int:

x=set(nums)

return (sorted(x))

# Nested list of student's info in a Science Olympiad

# List elements: (Student's Name, Marks out of 100 , Age)

participant\_list = [

('Alison', 50, 18),

('Terence', 75, 12),

('David', 75, 20),

('Jimmy', 90, 22),

('John', 45, 12)

]

sorted\_list = sorted(participant\_list, key=lambda item: (100-item[1], item[2]))

print(sorted\_list)

**Output**

[('Jimmy', 90, 22), ('Terence', 75, 12), ('David', 75, 20), ('Alison', 50, 18), ('John', 45, 12)]

----------------------------------------------------------------------------------------------------

Dict = dict(

{5: 'Geeks', 6: 'For', 8:'Geeks'}

)

for k,idx in enumerate(Dict):

print(k,idx,Dict[idx])

0 5 Geeks

1 6 For

2 8 Geeks

----------------------------------------------------------------------------

**Method #1:** Using max() function

|  |
| --- |
| # Python code to find key with Maximum value in Dictionary  # Dictionary Initialization  Tv = {'BreakingBad':100, 'GameOfThrones':1292, 'TMKUC' : 88}  Keymax = max(Tv, key=Tv.get)  print(Keymax)  o/p: |

GameOfThrones

----------------------------------------------------------------------------------------------------

string ="abcdcaf"

d=dict()

for i in range(len(string)):

if string[i] not in d:

d[string[i]]=1

else:

d[string[i]]+=1

value=""

for i,j in enumerate(d):

if d[j]==1:

print(string.index(j))

print("ok")

print(j,d[j])

value=j

break

print(string.index(value))

----------------------------------------------------------------------------------------------------

characters = "Bste!hetsi ogEAxpelrt x "

document = "AlgoExpert is the Best!"

res1=''.join(sorted(characters))

res2=''.join(sorted(document))

# print(res1.strip())

# print(res2.strip())

n = len(res1) if len(res1)<len(res1) else len(res2)

print(len(res1)," ",len(res2))

print(n)

c=0

for i in range(n):

if res1[i] == res2[i]:

c+=1

|  |
| --- |
| # Python code to illustrate the working of strip()  string = '   Geeks for Geeks   '    # Leading spaces are removed  print(string.strip())    # Geeks is removed  print(string.strip('   Geeks'))    # Not removed since the spaces do not match  print(string.strip('Geeks')) |

**Output :**

Geeks for Geeks

for

Geeks for Geeks

---------------------------------------------------------------------

Finding a value in or number in binary tree:

class Node :

def \_\_init\_\_(self,data):

self.data=data

self.left=None

self.right=None

def inorder(node):

if node is None:

return

inorder(node.left)

print(node.data,end=" ")

inorder(node.right)

def finding(node,target):

if node.data==target:

print("got")

print(node.data)

if node is None :

return 0

elif target>node.data:

return finding(node.right,target)

elif target<node.data:

return finding(node.left,target)

root=Node(10)

root.left=Node(5)

root.right=Node(15)

root.left.left =Node(2)

root.left.left.left =Node(1)

root.left.right=Node(5)

root.right.right=Node(22)

root.right.left=Node(13)

root.right.left.right=Node(14)

inorder(root)

finding(root,2)

**Problem Statement**

Alex works at a clothing store. There is a large pile of socks that must be paired by color for sale. Given an array of integers representing the color of each sock, determine how many pairs of socks with matching colors there are.

For example, there are **n=7** socks with colors **ar = {1,2,1,2,1,3,2}**. There is one pair of color **1** and one of color **2**. There are three odd socks left, one of each color. The number of pairs is **2**.

a=[1,1,2,2,3,3,5,4,8,6,6,9,9]

a.sort()

print(a)

a1=[]

i=0

while i < len(a)-1:

if a[i] == a[i+1]:

a1.append(a[i])

i+=1

i+=1

print(a1)

**Problem Statement**

A left rotation operation on an array shifts each of the array’s elements unit to the left. For example, if 2 left rotations are performed on array [1, 2, 3, 4, 5], then the array would become [3, 4, 5, 1, 2].

Given an array of integers and a number, , perform left rotations on the array. Return the updated array to be printed as a single line of space-separated integers.

a=[1,2,3,4,5]

n=2

i=0

while n > 0 :

a.append(a[0])

a.remove(a[0])

i+=1

n-=1

print(a)

**Example2:**

**input1:** 9  
**input2:** {15, 27, 14, 38, 63, 55, 46, 65, 85}

**Output:** 3  
**Explanation:**The longest decreasing subsequence is of length 3, i.e. [63, 55, 46]

a=[1,3,2,5,2,3,4,10,9,8,7,6]

count=0

greater = 0

for i in range(len(a)-1):

if a[i] > a[i+1] :

count += 1

print("in first block : ",a[i])

print("count is : ",count)

print("conparing : ",a[i],a[i+1])

else:

print("greater is : ",greater)

if count >= greater:

greater = count

print("count is : ",count)

print("in second block : ",a[i])

print("conparing : ",a[i],a[i+1])

count = 0

print("\n")

if count > greater :

greater = count

print(greater+1)

Given an array arr[] of integers, find out the maximum difference between any two elements such that the larger element appears after the smaller number.

**Examples :**  
**Input :** arr = {2, 3, 10, 6, 4, 8, 1}  
**Output :** 8  
**Explanation :** The maximum difference is between 10 and 2.

**Input :** arr = {7, 9, 5, 6, 3, 2}  
**Output** : 2  
**Explanation** : The maximum difference is between 9 and 7.

a=[7, 2, 9, 5, 6, 3, 2]

num=0

maxNum = max(a)

idx = a.index(maxNum)

small=maxNum

for i in range(idx):

if a[i] < small:

small = a[i]

print("smallest is : ",small)

print(abs(maxNum-small))

Given an array of integers (both odd and even), the task is to arrange them in such a way that odd and even values come in alternate fashion in non-decreasing order(ascending) respectively.

* If the smallest value is Even then we have to print Even-Odd pattern.
* If the smallest value is Odd then we have to print Odd-Even pattern.

Note: No. of odd elements must be equal to No. of even elements in the input array.

**Examples:**

**Input:**  arr[] = {1, 3, 2, 5, 4, 7, 10}

**Output**: 1, 2, 3, 4, 5, 10, 7

Smallest value is 1(Odd) so output will be Odd-Even pattern.

**Input:** arr[] = {9, 8, 13, 2, 19, 14}

**Output:** 2, 9, 8, 13, 14, 19

a=[9, 8, 13, 2, 19, 14]

a.sort()

b=[]

c=[]

for i in range(len(a)):

if a[i] % 2 == 0 :

b.append(a[i])

else:

c.append(a[i])

index = 0

i = 0

j = 0

flag = False

# Set flag to true if first element is even

if (a[0] % 2 == 0):

flag = True

while index < len(a) :

if flag == True and i<len(b):

a[index] = b[i]

index += 1

i += 1

flag = ~flag

elif j < len(c):

a[index] = c[j]

index += 1

j += 1

flag = ~flag

print(a)

Given a string s, the task is to find out the minimum number of adjacent swaps required to make a string is palindrome. If it is not possible, then return -1.

**Examples:**  
**Input:** aabcb  
**Output:** 3

**Explanation:**  
After 1st swap: abacb  
After 2nd swap: abcab  
After 3rd swap: abcba

**Input:** adbcdbad  
**Output:** -1

s="aabcb"

x = list(s)

for i in range(len(x)-1):

x[i],x[i+1] = x[i+1],x[i]

print(x)

**Dynamic programming :**

**Longest Palindromic Substring:**

s="abcba"

i=0

n = len(s)

a = [[0 for x in range(n)] for y in range(n)]

while i < n :

a[i][i] = True

i+=1

i = 0

j = i+1

while i < n-1:

if s[i] == s[j]:

a[i][j] = True

i+=1

j+=1

k = 3

while k <= n :

i=0

j = k - 1

while j < n:

if s[i] == s[j] and a[i+1][j-1] == True:

a[i][j] = True

i+=1

j+=1

k +=1

for r in a:

for c in r:

print(c,end = " ")

print()

Q] . reverse words in a string python .

a="hello krishna nakkum"

b = a.split(" ")

print(b)

c = " "

c = c.join(reversed(b))

print(c)

['hello', 'krishna', 'nakkum']

nakkum krishna hello

ascii :

A= 65

A = 97

print(ord('A')+1)

66

Q . swap adjacent

a=[1,2,3,4,5,6,7]

for i in range(0,len(a),2):

print(i)

if i == len(a)-1:

break

a[i],a[i+1] = a[i+1],a[i]

print(a)

**$python3 main.py**  
0

2

4

6

[2, 1, 4, 3, 6, 5, 7]

Q add numbers till int is less than or equal to z:

a=12345454

T=False

sum1=0

while T == False:

sum1=0

while a>0:

rem = a%10;

sum1 = sum1 + rem

a = a//10

if sum1 < 28 :

T = True

else:

a = int(sum1)

print(chr(96+sum1))

k

Q . :

def Check(a,target):

a.sort()

print(a)

arr=[]

start=0

end=len(a)-1

while start < end :

current=a[start]+a[end]

if current==target:

arr.append([a[start],a[end]])

start+=1

end-=1

elif current > target :

end-=1

elif current < target :

start+=1

return arr

print(Check([3, 5, 2, -4, 8, 11,10,-3],7))

**output:**

[-4, -3, 2, 3, 5, 8, 10, 11]

[[-4, 11], [-3, 10], [2, 5]]