```
*
**
***
****
*****
******
```

Make this pattern using nested while

In [9]:

```
i = 1
while i <= 5:
    j = 1
    while j<=i:
        print("*", end = " ")
        j+=1
    print()
    i+=1
    #print("\n")
```

```
*
* *
* * *
* * * *
* * * * *
```

Make this pattern using nested for

In [13]:

```
for i in range(1,7,1):
    for j in range(1,i,1):
        print("*", end = " ")
    print()
```

```
*
* *
* * *
* * * *
* * * * *
```

take user input and make star pattern

In [14]:

```
n = int(input("Enter the number of rows"))
for i in range(1,n+1):
    for j in range(1,i+1):
        print("*", end = " ")
    print()  #can use both print() and print("\n")
```

```
Enter the number of rows5
*
* *
* * *
* * * *
* * * * *
```

```
******
*****
****
***
**
*

```

make this pattern using nested for

In [15]:

```
n = int(input("enter the number of rows"))
for i in range(n,0,-1):
    for j in range(i,0,-1):
        print("*", end = " ")
    print()

```

```
enter the number of rows6
* * * * * *
* * * * *
* * * *
* * *
* *
*
```

Make this pattern using nested while

In [16]:

```
i = 5
while i>=1:
    j = 1
    while j<=i:
        print("*", end = " ")
        j+=1
    print()
    i-=1
```

```
* * * * *
* * * *
* * *
* *
*
```

```
----*
---**
--***
-****
*****
```

Make this pattern using nested while

In [17]:

```
i = 1
while i<=5:
    j = 1
    while j<=5-i:
        print(" ", end = " ")
        j+=1
    k = 1
    while k<=i:
        print("*", end = " ")
        k+=1
    print()
    i+=1
```

```
        *
      * *
    * * *
  * * * *
* * * * *
```

Make this pattern after taking user input using nested for

In [18]:

```
a = int(input("Enter the number of rows"))
for i in range(1,a+1):
    for j in range(i,a+1):
        print(" ", end = " ")
    for j in range(1,i+1):
        print("*", end = " ")
    print()
```

```
Enter the number of rows5
          *
        * *
      * * *
    * * * *
  * * * * *
```

```
*****
-****
--***
---**
----*
```

Make this loop using nested while

In [20]:

```
i = 5
while i>=1:
    j = 1
    while j<=5-i:
        print(" ", end =" ")
        j+=1
    k = 1
    while k<=i:
        print("*", end = " ")
        k+=1
    print()
    i-=1
```

```
* * * * *
  * * * *
    * * *
      * *
        *
```

Take user input and make this pattern using nested for

In [22]:

```
n = int(input("Enter number of rows"))
for i in range(0,n):
    for j in range(i,n+1):
        print(" ", end=" ")
    for j in range(i+1,0,-1):
        print("*", end=" ")
    print()
```

```
Enter number of rows5
            *
          * *
        * * *
      * * * *
    * * * * *
```

```
1
22
333
4444
55555
```

Make this pattern using nested while

In [23]:

```
i=1
while i<=5:
    j=1
    while j<=i:
        print(i, end = " ")
        j+=1
    print()
    i+=1
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

Make this pattern using nested for

In [24]:

```
for i in range(1,6):
    for j in range(0,i):
        print(i, end = " ")
    print()
```

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
1
```

```
12
123
1234
12345
```

Make this pattern using nested while

In [25]:

```
i = 1
while i<=5:
    j = 1
    while j<=i:
        print(j, end=" ")
        j+=1
    print()
    i+=1
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Make this pattern using nested for

In [29]:

```
for i in range(1,6):
    for j in range(1,i+1):
        print(j, end = " ")
    print()
```

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
____*____
___***___
__*****__
_*******_
*********
```

Make this pattern using nested while

In [32]:

```
i=1
while i<=5:
    j = 1
    while j<=5-i:
        print(" ", end=" ")
        j+=1
    k=1
    while k<=i:
        print("*", end=" ")
        k+=1
    h = 1
    while h < i:
        print('*',end=" ")
        h+=1
    print()
    i+=1
```

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

Make this pattern using nested for

In [33]:

```
for i in range(1,6):
    for j in range(5,i,-1):
        print(" ", end = " ")
    for k in range(1, i+1):
        print("*", end=" ")
    for l in range(1,i):
        print("*", end=" ")
    print()
```

```
        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

```
54321
4321
321
21
1
```

Make this pattern using nested while

In [35]:

```
i = 5
while i>=1:
    j = 1
    while j<=i:
        print(i, end =" ")
        i-=1
    print()
i = 4
while i>=1:
    j = 1
    while j<=i:
        print(i, end =" ")
        i-=1
    print()
i = 3
while i>=1:
    j = 1
    while j<=i:
        print(i, end=" ")
        i-=1
    print()
i = 2
while i>=1:
    j = 1
    while j<=i:
        print(i, end=" ")
        i-=1
    print()
i = 1
while i>=1:
    j = 1
    while j<=i:
        print(i, end=" ")
        i-=1
    print()
```

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

```
*********
-*******-
--*****--
---***---
----*----
```

Make this pattern using nested while

In [37]:

```
i = 5
while i>=1:
    j = 1
    while j<=5-i:
        print(" ", end = " ")
        j+=1
    k = 1
    while k<=i:
        print("*", end=" ")
        k+=1
    l = 1
    while l<i:
        print("*", end=" ")
        l+=1
    print()
    i-=1
```

```
* * * * * * * * *
  * * * * * * *
    * * * * *
      * * *
        *
```

Make this pattern using nested for.

In [38]:

```
for i in range(5,0,-1):
    for j in range(1,6-i):
        print(" ", end = " ")
    for k in range(1, i+1):
        print("*", end = " ")
    for l in range(1,i):
        print("*", end = " ")
    print()
```

```
* * * * * * * * *
  * * * * * * *
    * * * * *
      * * *
        *
```

```
----1----
---222---
--33333--
-4444444-
555555555
```

Make this pattern using nested for

In [39]:

```
for i in range(1,6):
    for j in range(5-i,0,-1):
        print(" ", end = " ")
    for k in range(0,i):
        print(i, end = " ")
    for l in range(1,i):
        print(i, end = " ")
    print()
```

```
        1
      2 2 2
    3 3 3 3 3
  4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5
```

Make this pattern using nested while.

In [40]:

```
i=1
while i<=5:
    j = 1
    while j<=5-i:
        print(" ", end = " ")
        j+=1
    k = 1
    while k<=i:
        print(i, end=" ")
        k+=1
    l = 1
    while l<i:
        print(i, end = " ")
        l+=1
    print()
    i+=1
```

```
        1
      2 2 2
    3 3 3 3 3
  4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5
```

```
111111111
-2222222-
--33333--
---444---
----5----
```

Make this pattern using nested for

In [41]:

```
n = 5
for i in range(1, n+1):
    for j in range(0,i):
        print(" ", end = " ")
    for k in range(i, n+1):
        print(i, end = " ")
    for l in range(i,n):
        print(i, end = " ")
    print()
```

```
 1 1 1 1 1 1 1 1 1
   2 2 2 2 2 2 2
     3 3 3 3 3
       4 4 4
         5
```

Make this pattern using nested while

In [43]:

```
i = 5
while i>=1:
    j=1
    while j<=5-i:
        print(" ", end = " ")
        j+=1
    k = 1
    while k<=i:
        print(j, end = " ")
        k+=1
    l = 1
    while l<i:
        print(j, end=' ')
        l+=1
    print()
    i-=1
```

```
1 1 1 1 1 1 1 1 1
  2 2 2 2 2 2 2
    3 3 3 3 3
      4 4 4
        5
```

```
123456789
-1234567-
--12345--
---123---
----1----
```

Make this pattern using nested for

In [44]:

```
for i in range(5,0,-1):
    for j in range(0,5-i):
        print(" ", end = " ")
    for k in range(1,i+1):
        print(k, end = " ")
    for l in range(1,i):
        print(l+i, end = " ")
    print()
```

```
1 2 3 4 5 6 7 8 9
  1 2 3 4 5 6 7
    1 2 3 4 5
      1 2 3
        1
```

Make this pattern using nested while

In [46]:

```
i = 5
while i>=1:
    j = 1
    while j<=5-i:
        print(" ", end = " ")
        j+=1
    k = 1
    while k<=i:
        print(k, end=" ")
        k+=1
    l = 1
    while l<i:
        print(l+i, end = " ")
        l+=1
    print()
    i-=1
```

```
1 2 3 4 5 6 7 8 9
  1 2 3 4 5 6 7
    1 2 3 4 5
      1 2 3
        1
```

```
----1----
---123---
--12345--
-1234567-
123456789
```

Make this pattern using nested for

In [47]:

```
for i in range(1,6):
    for j in range(0,5-i):
        print(" ", end = " ")
    for k in range(1, i+1):
        print(k, end = " ")
    for l in range(1, i):
        print(i+l, end =" ")
    print()
```

```
        1
      1 2 3
    1 2 3 4 5
  1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9
```

Make this pattern using nested while.

In [48]:

```
i = 1
while i<=5:
    j = 1
    while j<=5-i:
        print(" ", end = " ")
        j+=1
    k = 1
    while k<=i:
        print(k, end = " ")
        k+=1
    l = 1
    while l<i:
        print(i+l, end=" ")
        l+=1
    print()
    i+=1
```

```
        1
      1 2 3
    1 2 3 4 5
  1 2 3 4 5 6 7
1 2 3 4 5 6 7 8 9
```

```
54321
4321
321
21
1
```

Make this pattern using nested for

In [49]:

```
for i in range(5,0,-1):
    for j in range(0,i):
        print(i-j, end = " ")
    print()
```

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

Make this pattern using nested while.

In [51]:

```
i = 5
while i>=1:
    j = 0
    while j<i:
        print(i-j,end=" ")
        j+=1
    print()
    i-=1
```

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

```
----1----
---333---
--55555--
-7777777-
999999999
```

Make this pattern using nested for

In [52]:

```
for i in range(1,6):
    for j in range(0,5-i):
        print(" ", end = " ")
    for k in range(1, i+1):
        print(2*i-1, end = " ")
    for l in range(1,i):
        print(2*i-1, end = " ")
    print()
```

```
        1
      3 3 3
    5 5 5 5 5
  7 7 7 7 7 7 7
9 9 9 9 9 9 9 9 9
```

Make this pattern using nested while

In [53]:

```
i = 1
while i<=5:
    j = 1
    while j<=5-i:
        print(" ", end = " ")
        j+=1
    k = 1
    while k<=i:
        print(2*i-1,end=" ")
        k+=1
    l = 1
    while l<i:
        print(2*i-1, end=" ")
        l+=1
    print()
    i+=1
```

```
        1
      3 3 3
    5 5 5 5 5
  7 7 7 7 7 7 7
9 9 9 9 9 9 9 9 9
```

```
999999999
-7777777-
--55555--
---333---
----1----
```

Make this pattern using nested for

In [54]:

```
for i in range(5,0,-1):
    for j in range(0,5-i):
        print(" ", end = " ")
    for k in range(1,i+1):
        print(2*i-1, end = " ")
    for l in range(1,i):
        print(2*i-1, end = " ")
    print()
```

```
9 9 9 9 9 9 9 9 9
  7 7 7 7 7 7 7
    5 5 5 5 5
      3 3 3
        1
```

Make this pattern using nested while.

In [56]:

```
i = 5
while i>=1:
    j = 1
    while j<=5-i:
        print(" ", end=" ")
        j+=1
    k = 1
    while k<=i:
        print(2*i-1, end= " ")
        k+=1
    l = 1
    while l<i:
        print(2*i-1, end=" ")
        l+=1
    print()
    i-=1
```

```
9 9 9 9 9 9 9 9 9
  7 7 7 7 7 7 7
    5 5 5 5 5
      3 3 3
        1
```

```
p
py
pyt
pyth
pytho
python
pytho
pyth
pyt
py
p
```

In [57]:

```
a = "python"
j = 1
while j<=6:
    b = a[0:j]
    print(b)
    j+=1
k = 5
while k>=0:
    c = a[0:k]
    print(c)
    k-=1
```

```
p
py
pyt
pyth
pytho
python
pytho
pyth
pyt
py
p
```

Take user input and print any string and find the vowel in the string

In [59]:

```
a = str(input("enter any word  "))
x = len(a)
vowel = 0
for i in range(0,x):
    if a[i] =="a" or a[i]=="e" or a[i]=="i" or a[i]=="o" or a[i]=="u":
        print(a[i], end =" ")
        vowel +=1
print()
print("there are", vowel, "vowels")
```

```
enter any word  Hello
e o
there are 2 vowels
```

Create a dictionary using user input

In [61]:

```
a = int(input("enter the length of the dictionary "))
d = {}
x = 0
while x<=a-1:
    key=input("enter the key ")
    value=input("enter the value ")
    d.update({key:value})
    x+=1
print(d)
```

```
enter the length of the dictionary 5
enter the key Aryaman
enter the value 21
enter the key Eishan
enter the value 26
enter the key Muskaan
enter the value 20
enter the key Priyanka
enter the value 30
enter the key Vansh
enter the value 17
{'Aryaman': '21', 'Eishan': '26', 'Muskaan': '20', 'Priyanka': '30',
'Vansh': '17'}
```

Given an integer "n" perform the following conditional actions:

if n is odd, print weird

if n is even and in inclusive range of 2 to 5, print not weird

if n is even and in inclusive range of 6 to 20, print weird

if n is even and greater than 20, print not weird

In [62]:

```
n = int(input("enter the number  "))
if n%2!=0:
    print("weird")
else:
    if n>=2 and n<=5:
        print("not weird")
    elif n>=6 and n<=20:
        print("weird")
    elif n>20:
        print("not weird")
```

```
enter the number  4
not weird
```

```
Given a list of integers and add an integer target return indices of the two
numbers such that they add up to get the target
input 1: nums = [2,7,11,15]  , target=9
```

```
output1: [0,1] (because num[0]+num[1]==9, we return [0,1])
```

In [15]:

```
a = [2,7,11,15]
l = len(a)
x = 0
target = 9
for i in range(l):
    for j in range(l):
        if a[i] + a[j] == target:
            break
            print

```

```
 current no=0 previous no=0
 current no=1 previous no=0
 current no=2 previous no=1
 current no=3 previous no=2
 current no=4 previous no=3
 current no=5 previous no=4
 current no=6 previous no=5
 current no=7 previous no=6
 current no=8 previous no=7
 current no=9 previous no=8
```

In [67]:

```
for i in range(10):
    if i == 0:
        print("current no=",i," ","previous no=",i)
    elif 0<i<10:
        print("current no=",i," ","previous no=",i-1)
```

```
current no= 0   previous no= 0
current no= 1   previous no= 0
current no= 2   previous no= 1
current no= 3   previous no= 2
current no= 4   previous no= 3
current no= 5   previous no= 4
current no= 6   previous no= 5
current no= 7   previous no= 6
current no= 8   previous no= 7
current no= 9   previous no= 8
```

Print the even places letters of the word 'pynative'

In [68]:

```
word = "pynative"
l = len(word)
for i in range(0,l,2):
    print(word[i])
```

```
p
n
t
v
```

```
Write a function ton return True if the first and last number of the
given list is same if the numbers are different return False

given:
l1 = [10,20,30,40,10]
l2 = [75,65,55,32,45]
```

In [69]:

```
n = int(input("enter the length "))
a=[]
for i in range(n):
    x = int(input("enter the numbers "))
    a.append(x)
print(a)
if a[0]==a[-1]:
    print("True")
else:
    print("False")
```

```
enter the length 5
enter the numbers 10
enter the numbers 20
enter the numbers 30
enter the numbers 40
enter the numbers 10
[10, 20, 30, 40, 10]
True
```

```
print a dict by user input then {2:56,1:2,5:12,4:24,6:18,3:323}
keys and values are sorted in numerical order by keys
it should be arranged in this way -- (1:2),(2:56),(3:323),etc..

```

In [70]:

```
a = {}
b = []
n = int(input("enter the length "))
for i in range(n):
    keys = int(input("Enter the keys "))
    values = int(input("Enter the values "))
    a.update({keys : values})
    b.append([keys,values])
print(a)
print(b)
result = sorted(b)
c = dict(result)
print(c)
```

```
enter the length 6
Enter the keys 2
Enter the values 56
Enter the keys 1
Enter the values 2
Enter the keys 5
Enter the values 12
Enter the keys 4
Enter the values 24
Enter the keys 6
Enter the values 18
Enter the keys 3
Enter the values 323
{2: 56, 1: 2, 5: 12, 4: 24, 6: 18, 3: 323}
[[2, 56], [1, 2], [5, 12], [4, 24], [6, 18], [3, 323]]
{1: 2, 2: 56, 3: 323, 4: 24, 5: 12, 6: 18}
```

```
'''
Create a new list from two lists using the following condition

Given two lists of numbers, write a program to create a new list
such that the new list should contain odd numbers from the first list
and even numbers from the second list.

Given:
list1 = [10,20,25,30,35]
list2 = [40,45,60,75,90]
Expected Output :
result list = [25,35,40,60,90]'''
```

In [71]:

```
list1 = [10,20,25,30,35]
list2 = [40,45,60,75,90]
result_list = []
for i in list1:
    if i%2!=0:
        result_list.append(i)
for j in list2:
    if j%2==0:
        result_list.append(j)
print(result_list)
```

```
[25, 35, 40, 60, 90]
```

```
'''
lets learn about list comprehensions you are gievn three integers x,y,and z
representing the dimensions of cuboid along with an integer "n"
print a list of all possible coordinates given by (i,j,k) on a 3D grid
where the sum i+j+k is not equal to n
here, 0<=i<=x, i can only take values from 0 till x
      0<=j<=y, j can only take values from 0 till y
      0<=k<=z, k can only take values from 0 till z

input:
x = 1
y = 1
z = 2
n = 3
all permutations of [i,j,k] are:
[[0,0,0],[0,0,1],[0,0,2],[0,1,0],[0,1,1],[0,1,2],[1,0,0],[1,0,1],[1,0,2],etc..]

but here we have to print the list permutations of [i,j,k]
which is not equal to n so we will take only these arrays:
[[0,0,0],[0,0,1],[0,0,2],[0,1,0],[0,1,1],[1,0,0],[1,0,1],etc..]'''

```

In [72]:

```
a = []
n = 3
i = 0
j = 0
k = 0
for i in range(2):
    for j in range(2):
        for k in range(3):
            if (i+j+k) < n:
                b = [i,j,k]
                a.append(b)
print(a)
```

```
[[0, 0, 0], [0, 0, 1], [0, 0, 2], [0, 1, 0], [0, 1, 1], [1, 0, 0], [1,
0, 1], [1, 1, 0]]
```

n factorial

In [74]:

```
n = int(input("enter the value  "))
a = 1
while n>=1:
    a = n*a
    n-=1
print(a)
```

```
enter the value  5
120
```

```
'''
With a given integral number n, write a program to generate a dictionary that contains (i, i*i)
such that is an integral number between 1 and n (both included) and then the program should print the dictionary.
Suppose the input is supplied to the program: 8
Then, the output should be: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}'''
```

In [75]:

```
n = int(input("enter the integer  "))
dictionary = {}
for i in range(1,n+1):
    keys = i
    values = i*i
    dictionary.update({keys:values})
print(dictionary)
```

```
enter the integer  8
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

```
'''Write a program which accepts a sequence of comma-separated numbers from console and
generate a list and a tuple which contains every number.
Suppose the input is supplied to the program: 34, 67, 55, 33, 12, 98
Then, the output should be:
['34', '67', '55', '33', '12', '98']
('34', '67', '55', '33', '12', '98')'''

```

In [76]:

```
n = int(input("enter the length  "))
l = []
for i in range(n):
    item=int(input("enter the values"))
    l.append(item)
print(l)
t = tuple(l)
print(t)
```

```
enter the length  6
enter the values34
enter the values67
enter the values55
enter the values33
enter the values12
enter the values98
[34, 67, 55, 33, 12, 98]
(34, 67, 55, 33, 12, 98)
```

```
'''Write a program that accepts a comma separated sequence of words as input
and prints the words in a comma-separated sequence
after sorting them alphabetically.
Suppose the input is supplied to the program: without, hello, bag, world
Then, the output should be: bag, hello, without, world '''
```

In [77]:

```
n = int(input("enter the length  "))
l = []
for i in range(n):
    words=input("Enter the words  ")
    l.append(words)
l1= sorted(l)
print(l1)
```

```
enter the length  4
Enter the words  without
Enter the words  hello
Enter the words  bag
Enter the words  world
['bag', 'hello', 'without', 'world']
```

```
'''
Write a program, which will find all such numbers between 1000 and 3000 (both
included) such that
each digit of the number is an even number.
The numbers obtained should be printed in a comma-separated sequence on a
single line
'''
```

In [79]:

```
l = []
for i in range(1000,3001):
    j = str(i)
    if int(j[0])%2==0 and int(j[1])%2==0 and int(j[2])%2==0 and int(j[3])%2==0:
        l.append(j)
print(l)
```

```
['2000', '2002', '2004', '2006', '2008', '2020', '2022', '2024', '202
6', '2028', '2040', '2042', '2044', '2046', '2048', '2060', '2062', '2
064', '2066', '2068', '2080', '2082', '2084', '2086', '2088', '2200',
'2202', '2204', '2206', '2208', '2220', '2222', '2224', '2226', '222
8', '2240', '2242', '2244', '2246', '2248', '2260', '2262', '2264', '2
266', '2268', '2280', '2282', '2284', '2286', '2288', '2400', '2402',
'2404', '2406', '2408', '2420', '2422', '2424', '2426', '2428', '244
0', '2442', '2444', '2446', '2448', '2460', '2462', '2464', '2466', '2
468', '2480', '2482', '2484', '2486', '2488', '2600', '2602', '2604',
'2606', '2608', '2620', '2622', '2624', '2626', '2628', '2640', '264
2', '2644', '2646', '2648', '2660', '2662', '2664', '2666', '2668', '2
680', '2682', '2684', '2686', '2688', '2800', '2802', '2804', '2806',
'2808', '2820', '2822', '2824', '2826', '2828', '2840', '2842', '284
4', '2846', '2848', '2860', '2862', '2864', '2866', '2868', '2880', '2
882', '2884', '2886', '2888']
```

```
'''
Write a program that calculates and prints the value according to the given formula:
Q = Square root of [(2 * C * D)/H]
Following are the fixed values of C and H: C is 50. H is 30.
D is the variable whose values should be input to your program in a comma-separated sequence.
Let us assume the following comma separated input sequence is given to the program: 100,150,180
The output of the program should be: 18, 22, 24
Hints: If the output received is in decimal form, it should be rounded off to its nearest value
(for example, if the output received is 26.0, it should be printed as 26)
In case of input data being supplied to the question, it should be assumed to be a console input
'''
```

In [81]:

```python
from math import sqrt

def formula(D,C=50,H=30):
    Q = sqrt((2 * C * D)/H)
    return Q
l1 = []
l2 = []
n = int(input("enter the length  "))
for i in range(n):
    D = int(input("enter the value  "))
    l1.append(D)
    result = formula(D)
    result=round(result)
    l2.append(result)
print(l1)
print(l2)
```

```
enter the length  3
enter the value  100
enter the value  150
enter the value  180
[100, 150, 180]
[18, 22, 24]
```

```python
'''
Write a program that accepts a sequence of whitespace separated words as input and
prints the words after removing all duplicate words and sorting them alphanumerically.
Suppose the input is supplied to the program: hello world and practice makes perfect and hello world again
Then, the output should be: again and hello makes perfect practice world'''
```

In [83]:

```
a = []
n = int(input("enter the length  "))
for i in range(n):
    words = input("enter the words  ")
    a.append(words)
for j in a:
    count = 0
    for k in a:
        if j == k:
            count+=1
            if count>1:
                a.remove(j)
b = sorted(a)
#print(b)
string = " "
for x in b:
    string+=x
    print(x, end = " ")
```

```
enter the length  10
enter the words  hello
enter the words  world
enter the words  and
enter the words  practice
enter the words  makes
enter the words  perfect
enter the words  and
enter the words  hello
enter the words  world
enter the words  again
again and hello makes perfect practice world
```

```
'''
Write a program that accepts a sentence and calculate the number of letters and digits.
Suppose the input is supplied to the program: hello world! 123
Then, the output should be: LETTERS 10
 DIGITS 3 '''
```

In [84]:

```
a = input("Enter the sentence  ")
letter = 0
digit = 0
for i in a:
    if i.isdigit():
        digit+=1
    elif i.isalpha():
        letter+=1
print("DIGITS: ",digit," ","LETTERS: ",letter)
```

```
Enter the sentence  hello world! 123
DIGITS:  3   LETTERS:  10
```

```
'''
Write a program that accepts a sentence and calculate the number of upper case letters and lower case letters.
Suppose the input is supplied to the program: Hello world!
```

```
Then, the output should be: UPPER CASE 1
   LOWER CASE 9'''
```

In [86]:

```
a = input("enter the sentence.  ")
uppercase = 0
lowercase = 0
for i in a:
    if i.isupper():
        uppercase+=1
    elif i.islower():
        lowercase+=1
print("Upper Case: ",uppercase,"Lower Case: ",lowercase)
```

```
enter the sentence.  Hello world!
Upper Case:  1 Lower Case:  9
```

```
'''
Use a list comprehension to square each odd number in a list.
The list is input by a sequence of comma-separated numbers.
Suppose the input is supplied to the program: 1,2,3,4,5,6,7,8,9
Then, the output should be: 1, 3, 5, 7, 9
1,2,9,4,25,6,49,64,81'''
```

In [87]:

```
a = []
l_odd = []
l_squares = []
l = int(input("Enter the length "))
for i in range(l):
    numbers = int(input("Enter the numbers "))
    a.append(numbers)
    if numbers%2!=0:
        l_odd.append(numbers)
        a.remove(numbers)
        squares = numbers * numbers
        l_squares.append(squares)
a.extend(l_squares)
print(l_odd)
print(a)
```

```
Enter the length 9
Enter the numbers 1
Enter the numbers 2
Enter the numbers 3
Enter the numbers 4
Enter the numbers 5
Enter the numbers 6
Enter the numbers 7
Enter the numbers 8
Enter the numbers 9
[1, 3, 5, 7, 9]
[2, 4, 6, 8, 1, 9, 25, 49, 81]
```

```
'''
Write a Python program that accepts a list of integers and calculates the
length and the fifth element. Return true if the length of the list is
8 and the fifth element occurs thrice in the said list.
```

```
Input:
[19, 19, 15, 5, 5, 5, 1, 2]
Output:
True
Input:
[19, 15, 5, 7, 5, 5, 2]
Output:
False'''
```

In [88]:

```
n = int(input("Enter the length  "))
l = []
count = 0
for i in range(n):
    integers = int(input("Enter the integers  "))
    l.append(integers)
print(l)
for j in l:
    if l[4]==j:
        count+=1
if count==3 and n==8:
    print("True")
else:
    print("False")
```

```
Enter the length  8
Enter the integers  19
Enter the integers  15
Enter the integers  19
Enter the integers  5
Enter the integers  5
Enter the integers  5
Enter the integers  1
Enter the integers  2
[19, 15, 19, 5, 5, 5, 1, 2]
True
```

```
'''
Write a program that computes the value of a+aa+aaa+aaaa with a given
digit as the value of a. Suppose the input is supplied to the program: 9
Then, the output should be: 11106'''

```

In [89]:

```
a = int(input("enter the value  "))
result = a + (a*10 + a) + (a*100 + a*10 + a) + (a*1000 + a*100 + a*10 + a)
print(result)
```

```
enter the value  9
11106
```

```
'''Find the sum of the series upto n terms. Write a program to calculate the
sum of series up to n term.
For example, if n =5 the series will become
2 + 22 + 222 + 2222 + 22222 = 24690'''
```

In [1]:

```
x = int(input("enter the n: "))
y = 0
summ=0
a = int(input("enter the number: "))
for i in range(x):
    y=(y*10+1)
    print(y*a,end=" ")
    summ+=(y*a)
print("=",summ)
```

```
enter the n: 5
enter the number: 2
2 22 222 2222 22222 = 24690
```

```
'''
Write a Python program to find the length of a given list of non-empty strings.

Input:
['cat', 'car', 'fear', 'center']
Output:
[3, 3, 4, 6]

Input:
['cat', 'dog', 'shatter', 'donut', 'at', 'todo', '']
Output:
[3, 3, 7, 5, 2, 4, 0]'''
```

In [91]:

```
n = int(input("enter the length: "))
l = []
x = []
for i in range(n):
    words = input("enter the words: ")
    l.append(words)
print(l)
for i in l:
    x.append(len(i))
print(x)
```

```
enter the length: 4
enter the words: cat
enter the words: car
enter the words: fear
enter the words: center
['cat', 'car', 'fear', 'center']
[3, 3, 4, 6]
```

```
'''
We are making n stone piles! The first pile has n stones.
If n is even, then all piles have an even number of stones.
If n is odd, all piles have an odd number of stones.
Each pile must have more stones than the previous pile but as few as possible.
Write a Python program to find the number of stones in each pile.
Input: 2
Output:
[2, 4]
Input: 10
```

```
Output:
[10, 12, 14, 16, 18, 20, 22, 24, 26, 28]
Input: 3
Output:
[3, 5, 7]
Input: 17
Output:
[17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]'''

```

In [92]:

```
n = int(input("enter the number: "))
a = []
if n%2 == 0:
    for i in range(n,3*n):
        if i%2==0:
            a.append(i)
    print(a)
elif n%2!=0:
    for i in range(n,3*n):
        if i%2!=0:
            a.append(i)
    print(a)
```

```
enter the number: 10
[10, 12, 14, 16, 18, 20, 22, 24, 26, 28]
```

```
'''
Write a Python program to check a given list of integers where
the sum of the first i integers is i.
Input:
[0, 1, 2, 3, 4, 5]
Output:
False
Input:
[1, 1, 1, 1, 1, 1]
Output:
True
Input:
[2, 2, 2, 2, 2]
Output:
False'''
```

In [93]:

```
n = int(input("enter the length: "))
list_1 = []
add = 0
for i in range(n):
    integers = int(input("enter the integers: "))
    list_1.append(integers)
    add+=integers
print(list_1)
if add == len(list_1):
    print("True")
else:
    print("False")
```

```
enter the length: 6
enter the integers: 0
enter the integers: 1
enter the integers: 2
enter the integers: 3
enter the integers: 4
enter the integers: 5
[0, 1, 2, 3, 4, 5]
False
```

```
'''
Write a Python program to find strings in a given list starting with
a given prefix. Go to the editor
Input:
[( ca,('cat', 'car', 'fear', 'center'))]
Output:
['cat', 'car']
Input:
[(do,('cat', 'dog', 'shatter', 'donut', 'at', 'todo'))]
Output:
['dog', 'donut']'''

```

In [94]:

```
n = int(input("enter the length: "))
l = []
l1 = []
for i in range(n):
    words = str(input("enter the words: "))
    l.append(words)
print(l)
prefix = str(input("enter the prefix: "))
for j in l:
    if j[0] + j[1] == prefix:
        l1.append(j)
print(l1)
```

```
enter the length: 4
enter the words: cat
enter the words: car
enter the words: fear
enter the words: center
['cat', 'car', 'fear', 'center']
enter the prefix: ca
['cat', 'car']
```

```
input:[-1,2,0,-9,-4,3,7,5,6,2,-2,1]
output:4
Find the least missing positive number in this list
```

In [95]:

```
n = int(input("enter the length: "))
l = []
x = []
y = []
for i in range(n):
    num = int(input("enter the integers: "))
    l.append(num)
    l.sort()
print(l)
for j in l:
    if -j not in l:
        x.append(j)
#print(x)
for k in x:
    if k<0:
        y.append(k)
        y.sort(reverse = True)
print(-y[0])
```

```
enter the length: 12
enter the integers: -1
enter the integers: 2
enter the integers: 0
enter the integers: -9
enter the integers: -4
enter the integers: 3
enter the integers: 7
enter the integers: 5
enter the integers: 6
enter the integers: 2
enter the integers: -2
enter the integers: 1
[-9, -4, -2, -1, 0, 1, 2, 2, 3, 5, 6, 7]
4
```

```
'''
Write a Python program to determine the direction ('increasing' or
'decreasing')
of monotonic sequence numbers. Go to the editor
Input:
[1, 2, 3, 4, 5, 6]
Output:
Increasing.
Input:
[6, 5, 4, 3, 2, 1]
Output:
Decreasing.'''
```

In [96]:

```
n = int(input("enter the length: "))
l = []
for i in range(n):
    num = int(input("enter the numbers: "))
    l.append(num)
print(l)
if l[0]==min(l):
    print("Increasing")
elif l[0]==max(l):
    print("Decreasing")
```

```
enter the length: 6
enter the numbers: 6
enter the numbers: 5
enter the numbers: 4
enter the numbers: 3
enter the numbers: 2
enter the numbers: 1
[6, 5, 4, 3, 2, 1]
Decreasing
```

```
'''
Write a Python program to determine which triplets sum to zero from a given list of lists.
Input: [[1343532, -2920635, 332], [-27, 18, 9], [4, 0, -4], [2, 2, 2], [-20, 16, 4]]
Output:
[False, True, True, False, True]
Input: [[1, 2, -3], [-4, 0, 4], [0, 1, -5], [1, 1, 1], [-2, 4, -1]]
Output:
[True, True, False, False, False]'''
```

In [97]:

```
n = int(input("enter the length: "))
list_1=[]

for i in range(n):
    list_2=[]
    for j in range(3):
        num = int(input("enter the numbers: "))
        list_2.append(num)
    list_1.append(list_2)
print(list_1)
output=[]
for k in list_1:
    if k[0]+k[1]+k[2] == 0:
        result = "True"
    else:
        result = "False"
    output.append(result)
print(output)
```

```
enter the length: 5
enter the numbers: 1
enter the numbers: 2
enter the numbers: -3
enter the numbers: -4
enter the numbers: 0
enter the numbers: 4
enter the numbers: 0
enter the numbers: 1
enter the numbers: -5
enter the numbers: 1
enter the numbers: 1
enter the numbers: 1
enter the numbers: -2
enter the numbers: 4
enter the numbers: -1
[[1, 2, -3], [-4, 0, 4], [0, 1, -5], [1, 1, 1], [-2, 4, -1]]
['True', 'True', 'False', 'False', 'False']
```

```
'''
Write a Python program to sort numbers based on strings. Go to the editor
Input: six one four one two three
Output:
one two three four six
Input: six one four three two nine eight
Output:
one two three four six eight nine
Input: nine eight seven six five four three two one
Output:
one two three four five six seven eight nine'''
```

In [98]:

```
n = int(input("enter the length: "))
l1 = ["one","two","three","four","five","six","seven","eight","nine"]
l2 = [1,2,3,4,5,6,7,8,9]
z = zip(l1,l2)
l = list(z)
nums = []
#print(l)
for i in range(n):
    numbers = str(input("enter the strings: "))
    nums.append(numbers)
print(nums)
final = []
final1 = []
for j in nums:
    for a,b in l:
        if a == j:
            final.append(b)
#print(final)
final.sort()
for k in final:
    for x,y in l:
        if y == k:
            final1.append(x)
print(final1)
```

```
enter the length: 7
enter the strings: six
enter the strings: one
enter the strings: four
enter the strings: three
enter the strings: two
enter the strings: nine
enter the strings: eight
['six', 'one', 'four', 'three', 'two', 'nine', 'eight']
['one', 'two', 'three', 'four', 'six', 'eight', 'nine']
```

```
'''
Write a Python program to find the set of distinct characters in a given
string,
ignoring case. Go to the editor
Input: HELLO
Output:
['h', 'o', 'l', 'e']
Input: HelLo
Output:
['h', 'o', 'l', 'e']
Input: Ignoring case
Output:
['s', 'n', 'c', 'o', 'e', 'i', 'r', 'g', 'a', ' ']'''
```

In [102]:

```
words = str(input("enter the string: "))
x = words.lower()
l = []
for i in x:
    l.append(i)
#print(l)
for i in l:
    count = 0
    for j in l:
        if i == j:
            count+=1
            if count>1:
                l.remove(i)
print(l)
```

```
enter the string: Hello
['h', 'e', 'l', 'o']
```

```
'''
Write a Python program to find all words in a given string with n consonants.
Go to the editor
Input: this is our time
Output:
Number of consonants: 3
Words in the said string with 3 consonants:
['this']
Number of consonants: 2
Words in the said string with 2 consonants:
['time']
Number of consonants: 1
Words in the said string with 1 consonants:
['is', 'our']'''
```

In [103]:

```
x = []
vowels = ["a","e","i","o","u"]
n = int(input("enter the length of the list: "))
for i in range(n):
    s = str(input("enter the words: "))
    x.append(s)
print(x)
for i in range(n):
    for j in vowels:
        x[i] = x[i].replace(j,"")
#print(x)
for a in x:
    l = len(a)
    print(a,"has",l,"consonants")
```

```
enter the length of the list: 4
enter the words: this
enter the words: is
enter the words: our
enter the words: time
['this', 'is', 'our', 'time']
ths has 3 consonants
s has 1 consonants
r has 1 consonants
tm has 2 consonants
```

```
'''
Write a Python program to find the h-index, the largest positive number h such
that
h occurs in the sequence at least h times. If there is no such positive number
return h = -1. Go to the editor
Input:
[1, 2, 2, 3, 3, 4, 4, 4, 4]
Output:
4
Input:
[1, 2, 2, 3, 4, 5, 6]
Output:
2
Input:
[3, 1, 4, 17, 5, 17, 2, 1, 41, 32, 2, 5, 5, 5, 5]
Output:
5  '''
```

In [104]:

```
n = int(input("enter the length: "))
l = []
l1 = []
for i in range(n):
    num = int(input("enter positive numbers: "))
    l.append(num)
print(l)
for j in l:
    count = 0
    for k in l:
        if k == j:
            count += 1
            if count >=k:
                l1.append(k)
#print(l1)
if len(l1)>=1:
    print(max(l1))
elif len(l1) == 0:
    print(-1)
```

```
enter the length: 7
enter positive numbers: 1
enter positive numbers: 2
enter positive numbers: 2
enter positive numbers: 3
enter positive numbers: 4
enter positive numbers: 5
enter positive numbers: 6
[1, 2, 2, 3, 4, 5, 6]
2
```

```
'''
Write a Python program to find even-length words from a given list of words and
sort them by length. Go to the editor
Original list of words:
['Red', 'Black', 'White', 'Green', 'Pink', 'Orange']
Find the even-length words and sort them by length in the said list of words:
['Pink', 'Orange']
Original list of words:
['The', 'worm', 'ate', 'a', 'bird', 'imagine', 'that', '!', 'Absurd', '!!']
Find the even-length words and sort them by length in the said list of words:
['!!', 'bird', 'that', 'worm', 'Absurd']
'''
```

In [2]:

```
n = int(input("enter the length: "))
l = []
for i in range(n):
    words = input("enter the words: ")
    l.append(words)
print(l)
l1 = []
for j in l:
    if len(j)%2==0:
        l1.append(j)
#print(l1)
l2 = []
for k in l1:
    length_k = len(k)
    l2.append(length_k)
#print(l2)
z = zip(l1,l2)
d = dict(z)
#print(d)
sorted_d = sorted(d.items(), key=lambda x:x[1])
d1 = dict(sorted_d)
print(list(d1.keys()))
```

```
enter the length: 10
enter the words: The
enter the words: worm
enter the words: ate
enter the words: a
enter the words: bird
enter the words: imagine
enter the words: that
enter the words: !
enter the words: Absurd
enter the words: !!
['The', 'worm', 'ate', 'a', 'bird', 'imagine', 'that', '!', 'Absurd',
'!!']
['!!', 'worm', 'bird', 'that', 'Absurd']
```

```
'''
Write a Python program to find the string consisting of all the words whose
lengths
are prime numbers.
Input:
The quick brown fox jumps over the lazy dog.
Output:
The quick brown fox jumps the'''
```

In [7]:

```
n = int(input("enter the length: "))
l = []
for i in range(n):
    words = input("enter the words: ")
    l.append(words)
#print(l)
for x in l:
    #print(x, end = " ")
    l1 = []
for j in l:
    y = 2
    len1 = len(j)
    if len1<y and len1 == y:
        break
    elif len1>y:
        while y<len1:
            if len1%y==0:
                break
            y+=1
        else:
            l1.append(j)
#print(l1)
for k in l1:
    print(k, end = " ")
```

```
enter the length: 9
enter the words: The
enter the words: quick
enter the words: brown
enter the words: fox
enter the words: jumps
enter the words: over
enter the words: the
enter the words: lazy
enter the words: dog.
The quick brown fox jumps the
```

```
'''
Check whether a string is a palangram(a sentence that contains every letter in
the english alphabet. Ex: The quick brown fox jumps over the lazy dog) or not.
'''
```

In [8]:

```
a = str(input("enter the sentence: "))
a = a.lower()
s = set(a)
count = 0
for i in s:
    if i.isalpha() == True:
        count+=1
#print(count)
if count == 26:
    print("the string is a palangram")
else:
    print("the string is not a palangram")
```

```
enter the sentence: The quick brown fox jumps over the lazy dog
the string is a palangram
```

```
'''
Write a program to check whether a set is a subset or not
subset - all elements of set-A are in set-B so set-A is a subset of set-B'''

```

In [9]:

```
A = {3,4,5,6}
B = {4,5,0}
count = 0
for i in A:
    for j in B:
        if i == j:
            count+=1
            break
if count == len(B):
    print("B is a subset a A")
else:
    print("B is not a subset of A")
```

```
B is not a subset of A
```

```
'''
Write a Python program to reverse the case of all strings. For those strings,
which contain no letters, reverse the strings. Go to the editor
Original list:
['cat', 'catatatatctsa', 'abcdefhijklmnop', '124259239185125', '', 'foo',
'unique']
Reverse the case of all strings. For those strings which contain no letters,
reverse
the strings:
['CAT', 'CATATATATCTSA', 'ABCDEFHIJKLMNOP', '521581932952421', '', 'FOO',
'UNIQUE']
Original list:
['Green', 'Red', 'Orange', 'Yellow', '', 'White']
Reverse the case of all strings. For those strings which contain no letters,
reverse the strings:
['gREEN', 'rED', 'oRANGE', 'yELLOW', '', 'wHITE']
Original list:
['Hello', '!@#', '!@#$', '123#@!']
Reverse the case of all strings. For those strings which contain no letters,
reverse the strings:
```

```
['hELLO', '#@!', '$#@!', '!@#321'] '''
```

In [10]:

```
n = int(input("Enter the length: "))
l = []
l1 = []
for i in range(n):
    a = (input("Enter the strings: "))
    l.append(a)
print(l)
for j in l:
    x = len(j)
    if j.isalpha():
        j = j.swapcase()
        l1.append(j)
    else:
        l1.append(j[x-1::-1])
print(l1)
```

```
Enter the length: 4
Enter the strings: Hello
Enter the strings: !@#
Enter the strings: !@#$
Enter the strings: 123#@!
['Hello', '!@#', '!@#$', '123#@!']
['hELLO', '#@!', '$#@!', '!@#321']
```

```
'''
[-1,2,0,-9,-4,3,7,5,6,2,-2,1]
Find the least missing positive number in this list using set function'''
#using sets
```

In [11]:

```
n = int(input("enter the length: "))
l = []
for i in range(n):
    num = int(input("Enter the numbers: "))
    l.append(num)
print(l)
s = set(l)
s1 = sorted(s)
#print(s1)
s2 = set()
for j in s1:
    if j<0:
        if -j not in s1:
            s2.add(-j)
#print(s2)
print(min(s2))
```

```
enter the length: 12
Enter the numbers: -1
Enter the numbers: 2
Enter the numbers: 0
Enter the numbers: -9
Enter the numbers: -4
Enter the numbers: 3
Enter the numbers: 7
Enter the numbers: 5
Enter the numbers: 6
Enter the numbers: 2
Enter the numbers: -2
Enter the numbers: 1
[-1, 2, 0, -9, -4, 3, 7, 5, 6, 2, -2, 1]
4
```

```
'''Write a program that takes full name as input and displays the abbrevations
of the first and middle names except the last name which is displayed as it
is. Eg: Robert Brett Roser output :R.B.Roser.'''
```

In [12]:

```
first_name = input("enter first name: ")
middle_name = input("enter middle name: ")
last_name = input("enter last name: ")
print(first_name[0],middle_name[0],last_name)
```

```
enter first name: Robert
enter middle name: Brett
enter last name: Rosser
R B Rosser
```

```
'''
Write a Python program to add 'ing' at the end of a given string (length
should be at least 3). If the given string already ends with 'ing' then add
'ly'
instead. If the string length of the given string is less than 3, leave
it unchanged.
Sample String : 'abc'
Expected Result : 'abcing'
Sample String : 'string'
```

```
Expected Result : 'stringly'''
```

In [13]:

```
a = str(input("enter the string: "))
x = len(a)
if x<3:
    print(a)
elif x>=3:
    if a[-1] == "g" and a[-2] == "n" and a[-3] == "i":
        a+="ly"
        print(a)
    else:
        a+="ing"
        print(a)
```

```
enter the string: abc
abcing
```

```
'''
Write a Python function to convert a given string to all uppercase if it
contains at least 2 uppercase characters in the first 4 characters.'''

```

In [14]:

```
a = str(input("enter the string: "))
count = 0
for i in a[0:4]:
    if i.isupper():
        count+=1
if count>=2:
    print(a.upper())
else:
    print(a)
```

```
enter the string: HEllo
HELLO
```

```
'''
Write a Python program to print the index of the character in a string.
Sample string: deepanshu
Expected output:
Current character d position at 0
Current character e position at 1
Current character e position at 2
- - - - - - - - - - - - - - - - - -
Current character h position at 8
Current character u position at 9 '''
```

In [16]:

```
a = str(input("enter the string: "))
x = len(a)
for i in range(x):
    print("Current character",a[i],"position at",i)
```

```
enter the string: aryaman
Current character a position at 0
Current character r position at 1
Current character y position at 2
Current character a position at 3
Current character m position at 4
Current character a position at 5
Current character n position at 6
```

```
'''
Write a Python program to get the single digits in numbers sorted backwards
and converted into English words. Go to the editor
Input:
[1, 3, 4, 5, 11]
Output:
['five', 'four', 'three', 'one']
Input:
[27, 3, 8, 5, 1, 31]
Output:
['eight', 'five', 'three', 'one']'''
```

In [17]:

```
n = int(input("enter the length: "))
l = []
for i in range(n):
    num = int(input("enter the numbers: "))
    l.append(num)
print(l)
l1 = []
for j in l:
    if 0<=j<10:
        l1.append(j)
#print(l1)
l1.sort(reverse = True)
#print(l1)
a = [1,2,3,4,5,6,7,8,9]
b = ["one","two","three","four","five","six","seven","eight","nine"]
z = zip(a,b)
l2 = list(z)
#print(l2)
l3 = []
for x in l1:
    for a,b in l2:
        if x == a:
            l3.append(b)
print(l3)
```

```
enter the length: 5
enter the numbers: 1
enter the numbers: 3
enter the numbers: 4
enter the numbers: 5
enter the numbers: 11
[1, 3, 4, 5, 11]
['five', 'four', 'three', 'one']
```

```
'''
Write a Python program to find the index of the largest prime in the list and
the sum of its digits. Go to the editor
Input: [3, 7, 4]
Output:
[1, 7]
Input: [3, 11, 7, 17, 19, 4]
Output:
[4, 10]
Input: [23, 17, 201, 14, 10473, 43225, 421, 423, 11, 10, 2022, 342157]
Output:
[6, 7]'''
```

In [18]:

```
n = int(input("enter the length: "))
l = []
for i in range(n):
    nums = int(input("enter the numbers: "))
    l.append(nums)
print(l)
l1 = []
count = 0
for j in l:
    a = 2
    if j<a:
        count+=1
        if count == len(l):
            print("no prime numbers found in the list.")
    elif j == a:
        l1.append(j)
    elif j>a:
        while a<j:
            if j%a==0:
                break
            a+=1
        else:
            l1.append(j)
#print(l1)
l2 = []
for n in range(len(l)):
    if max(l1) == l[n]:
        l2.append(n)
x = max(l1)
x = str(x)
summ = 0
for k in x:
    summ+=int(k)
l2.append(summ)
print(l2)
```

```
enter the length: 6
enter the numbers: 3
enter the numbers: 11
enter the numbers: 7
enter the numbers: 17
enter the numbers: 19
enter the numbers: 4
[3, 11, 7, 17, 19, 4]
[4, 10]
```

```
'''
Write a Python program to find a string consisting of space-separated
characters with given counts. Go to the editor
Input: {'f': 1, 'o': 2}
Output:
f o o
Input: {'a': 1, 'b': 1, 'c': 1}
Output:
a b c'''
```

In [11]:

```
n = int(input("enter the length: "))
d = {}
for i in range(n):
    keys = input("enter the keys: ")
    values = int(input("enter the values: "))
    d.update({keys:values})
print(d)
z = zip(d.keys(),d.values())
l1 = list(z)
#print(l1)
for a,b in l1:
    c = a * b
    for x in c:
        print(x, end = " ")
```

```
enter the length: 2
enter the keys: f
enter the values: 1
enter the keys: o
enter the values: 2
{'f': 1, 'o': 2}
f o o
```

In [ ]:

```

```