#dictionary

#key-value pair  
#duplicates values are allowed, but keys cannot be   
#duplicated key act as a index  
#homo/heterogenous  
#insertion order is not preserved  
#dict are mutable  
#no indexing and slicing

#empty dictionary

#d={}  
#d=dict()  
#print(type(d))

#d={100:"python",200:"java",300:"c++"}  
#print(d[100])

#update

#d[key]=value

#if key is not avaialble then new entry will be added  
#if available new value will be updated

#d={100:"python",200:"java",300:"c++"}  
##d[400]="spark"  
#d[100]="DBMS"  
#print(d)

#delete- del d[key]

#d={100:"python",200:"java",300:"c++"}  
##print(d)  
#del d  
#print(d)  
#  
#d.clear()

#len(),

#get()- to get the value associated with the key  
#if key is available returns the corresponding value  
#otherwise returns None  
#d={100:"python",200:"java",300:"c++"}  
##print(d.get(100))  
#  
#print(d.get(400))

#pop()-- d.pop(key)  
#d={100:"python",200:"java",300:"c++"}  
#print(d.pop(400))

#popitem()-removes arbitary item(key-value)from the dic  
#d={100:"python",200:"java",300:"c++"}  
#print(d)  
#print(d.popitem())  
#print(d)

#values-return all values asscoiated with dict  
#d={100:"python",200:"java",300:"c++"}  
#print(d.values())  
#for v in d.values():  
# print(v)

#items()- it returns list of tuples represeting K-V pair  
#d={100:"python",200:"java",300:"c++"}  
#for k,v in d.items():  
# print(k,"---",v)

#copy()-cloned copy

#setdefault  
#d={100:"python",200:"java",300:"c++"}  
#print(d.setdefault(400,"DBMS"))  
#print(d)

#Modules  
#a group of functions,vraibles,classes saved to a file.  
#every .py file acts as a module

demo1.py

​

[10:37 am] Poornamathi (Guest)

a=1000

def add(x,y):     print(" the sum is",x+y)

#test.py  
#import modulename  
#print(modulename.vairbalename)  
#print(modulename.function())

#import demo1  
#print(demo1.a)  
#demo1.add(10,20)

#from demo1 import a,add#importing specific module  
#print(a)  
#add(100,200)

#from demo1 import \* #import all modules  
#  
#import module1,module2,module3  
#  
#x=100  
#y=200  
#def f1():  
# print("welcome")  
#print(dir())# print all mmebers of current module

#for every module at the tome of execution python   
#intepreter will add some special properties automatically  
#for internal use

#import demo1  
#print(dir(demo1))  
#print(\_\_file\_\_)  
#print(\_\_package\_\_)  
#print(\_\_cached\_\_)

#from math import \*  
#print(sqrt(4))  
##ceil,floor,abs  
#from random import \* #generate some float values b/w 0 and 1  
##print 0< x < 1  
#for i in range(10):  
# print(random())

**Arrange Names**

Teachers of Horizon Academy Public School collect the names of those students who are going to participate in the Mathematics exhibition, that is to be held in a week's time. They have collected all the names and now, they want to store all the names into a system, in the descending order of the length of the names. This means, the longest name should get stored first, followed by the name that is shorter than the previous, and so on. Can you help the teachers to perform this task easily by creating a program in Python?

Note: The number of names specified must be positive, else the program should display the message "Invalid Input" terminate the program.

Input format:

Input consists of an integer that corresponds to the number of names followed by the names.

Output Format:

Print the name list sorted by the names' length, if name length are equal sort based on alphabetical order in descending order as shown in the sample input and output.

[10:35 am] Poornamathi (Guest)

Sample Input 1:

Enter the number of names :

5

Enter the names:

William

James

Ella

Lily

Jackson

Sample Output 1:

The sorted name list is:

William

Jackson

James

Lily

Ella

Sample Input 2:

Enter the number of names:

-1

Sample Output 2:

Invalid Input

Sample Input 3:

Enter the number of names :

3

Enter the names:

Lily

Jack

Lucy

Sample Output 3:

The sorted name list is:

Lily

Lucy

Jack

​

## Rhythm Composer

Imagine D Iman, a famous music director to be composing a pop album. To create a unique composition, he decides to play only those keys of his instrument (consider every key has a number associated with it) which are divisible by one and itself (i.e. prime numbers). You may help him by writing a program in Python to print all the numbers that represent the keys he will need for his new composition - given the interval [start,end] (start and end inclusive) within which he chooses the keys. Write a user defined function to find out the key with prime numbers.

Note:

* If start or end number is negative, then display the message "Invalid range".
* If there is not prime numbers between the starting and ending range, then display the message "There is no prime numbers in this range".
* If the start and end numbers are same, then display the message "There is no prime numbers in this range".
* If the start number is greater than end number, then display the message "Invalid range".
* The function definition should be : find\_prime(start,end) - This function should take start and end number as parameters and return the prime numbers as a list.

Refer the sample input and output statements for more clarifications.

Sample Input 1:

2

11

Sample Output 1:

2 3 5 7 11

Sample Input 2:

3

-10

Sample Output 2:

Invalid range

Sample Input 3:

0

1

Sample Output 3:

There is no prime numbers in this range

Sample Input 4:

1

1

Sample Output 4:

There is no prime numbers in this range

Sample Input 5:

1

11

Sample Output 5:

2 3 5 7 11