Given a string, you have to find the first n most frequent characters in it.

You have to print these n letters in alphabetically sorted order.

The input will contain two lines, the first line will contain a string and the second line will contain the letter n.

The output should be a list of n most frequent letters in alphabetically sorted order.

**Note:**If there are two letters with the same frequency, then the alphabetically preceding alphabet should be picked first. (For example, in "aabbccc", if n=2, then output list would have c and a.)

Sample Input:

ddddaacccb

3

Sample Output:

['a', 'c', 'd']

In the above example, the order of frequencies is : d>c>a>b

Here, d,c and a are three most frequent characters which are displayed in alphabetically sorted order.

Answer:

Approach1

import pandas as pd

string=input()

n=int(input())

#Getting unique items from string by converting it into Set

set\_string = set(string)

#Declared dataframe to store the key value pairs of letter-count

df = pd.DataFrame(columns = ['letter','count'])

#Populating the dataframe

for i in set\_string:

cnt = string.count(i)

df = df.append({'letter': i, 'count': cnt}, ignore\_index = True)

string.replace(i,'')

#Sorting dataframe on count first and then on letters

df.sort\_values([ 'count', 'letter'], ascending=[False, True], inplace =True)

#Getting first n entries and returning sorted array

print(sorted(list(df.letter)[:n]))

Approach 2:

string=input()

n=int(input())

import collections

out=[collections.Counter(string).most\_common(i+1)[i][0] for i in range(n)]

out.sort()

print(out)