`Problem 4` : Write a program that will take 2 numbers as input and prints the LCM and HCF of those 2 numbers

In [20]:

```
num1=int(input("Provide first num for LCM and HCF: "))
num2=int(input("Provide second num for LCM and HCF: "))

import math

lcm=math.lcm(num1,num2)
hcf=math.gcd(num1,num2)

print(lcm)
print(hcf)
```

```
Provide first num for LCM and HCF: 22
Provide second num for LCM and HCF: 10
110
2
```

Problem 5: Create Short Form from initial character

Given a string create short form ofthe string from Initial character. Short form should be capitalised.

Example:

`Input:`

```
Data science mentorship program
```

`Output:`

```
DSMP
```

In [9]:

```
sentence="Data science mentorship program"

splitword=sentence.split()
print(splitword) # just to visualize how it looks so that we can understand clearly
```

```
['Data', 'science', 'mentorship', 'program']
```

In [11]:

```
print(type(splitword))
```

```
<class 'list'>
```

In [12]:

```
#here above we convert sentence to list now, we can take out first alphabet of every word
for i in splitword:
    print(i[0])
```

```
D
s
m
p
```

In [13]:

```
#now we need every word capital and in same sentence
sentence="Data science mentorship program"
splitword=sentence.split()

for i in splitword:
    print(i[0].upper(), end=" ")
```

```
D S M P
```

`Problem 6` : Append second string in the middle of first string

`Input:`

```
campusx
data
```

`Output` :

```
camdatapusx
```

In [15]:

```
a="campusx"
b="data"
```

In [20]:

```
a1=a[0:3]
b1=a[3:7]
print(a1)
```

```
cam
```

In [21]:

```python
print(b1)
```

```
pusx
```

In [22]:

```python
#now lets concatenate:

c= a1+b+b1
print(c)
```

```
camdatapusx
```

`Problem 7`:Given string contains a combination of the lower and upper case letters. Write a program to arrange the characters of a string so that all lowercase letters should come first.

Given:

`str1 = PyNaTive`

Expected Output:

`yaivePNT`

In [27]:

```python
#think it as odd and even number (like differentaiation of even and odd)

capital=""
small=""

str1="PyNaTive"

for i in str1:
    if i.islower():
        small=small+i
    else:
        capital=capital+i

print(capital)
print(small)

print(small+capital)
```

```
PNT
yaive
yaivePNT
```

In [1]:

```python
#OR
#ABCeDfg
#lower lower together

lower=[]
capital=[]

a="ABcDeFgH"
for i in a:
    if i.islower():
        lower.append(i)
    else:
        capital.append(i)

print("".join(lower)+"".join(capital))
```

```
cegABDFH
```

`Problem 8:` Take a alphanumeric string input and print the sum and average of the digits that appear in the string, ignoring all other characters.

Input:

`hel12304every093`

Output:

```
    Sum: 22
    Avg: 2.75
```

In [32]:

```python
a="hel12304every093"
summation=0
average=0


for i in a:
    if i.isdigit():
        summation=summation+(int(i))

print(summation)
```

```
22
```

In [45]:

```
#it will work for sum but not for average
#so we need to use lists too.
```

In [47]:

```
a="hel12304every093"
digits=[]

for i in a:
    if i.isdigit():
        digits.append(int(i))


print(digits)
```

```
[1, 2, 3, 4, 0, 9, 3]
```

In [49]:

```
print(sum(digits))
print(sum(digits)/len(digits))
```

```
22
3.142857142857143
```

In [53]:

```
#OR we can do directly

import statistics

a="hel12304every093"
digits=[]
summation=0

for i in a:
    if i.isdigit():
        digits.append(int(i))
        summation=sum(digits)

print(summation)
print(statistics.mean(digits))
```

```
22
3.142857142857143
```

`Problem 9:` Removal of all characters from a string except integers

`Given:`

```
str1 = 'I am 25 years and 10 months old'
```

`Expected Output:`

```
2510
```

In [8]:

```
#its looks different but similar to previous one

str1 = 'I am 25 years and 10 months old'
num=[]

for i in str1:
    if i.isdigit():
        num.append(int(i))
print(num)
```

```
[2, 5, 1, 0]
```

`Problem 10`: Check whether the string is Symmetrical.

Statement: Given a string. the task is to check if the string is symmetrical or not. A string is said to be symmetrical if both the halves of the string are the same.

Example 1:

`Input`

khokho

`Output`

`The entered string is symmetrical`

```
a=input("enter you want to check whether it is symmetrical or not: ")
```

In [9]:

```
a= input("enter you want to check whether it is symmetrical or not: ")
n=len(a)//2

a1=a[0:n]
a2=a[n:]

print(a1)
print(a2)
```

```
enter you want to check whether it is symmetrical or not: khokho
kho
kho
```

In [10]:

```
#now lets move forward, we are going right
a= input("enter you want to check whether it is symmetrical or not: ")
n=len(a)//2

a1=a[0:n]
a2=a[n:]

if a1==a2:
    print("it is symmetrical")
else:
    print("it is not symmetrical")
```

```
enter you want to check whether it is symmetrical or not: khokho
it is symmetrical
```

`Problem 11` : Reverse words in a given String

Statement: We are given a string and we need to reverse words of a given string.

Example 1:

`Input:`

```
geeks quiz practice code
```

`Output:`

```
code practice quiz geeks
```

Example 2:

`Input:`

```
my name is laxmi
```

`Output:`

```
laxmi is name my
```

In [18]:

```
a="my name is laxmi"
b=a.split()
print(b)
```

```
['my', 'name', 'is', 'laxmi']
```

In [21]:

```
c= b[::-1]
print(c)
```

```
['laxmi', 'is', 'name', 'my']
```

In [23]:

```
" ".join(c)
```

Out[23]:

```
'laxmi is name my'
```

`Problem 12` : Find uncommon words from two Strings.

Statement: Given two sentences as strings **A** and **B**. The task is to return a list of all uncommon words. A word is uncommon if it appears exactly once in any one of the sentences, and does not appear in the other sentence. Note: A sentence is a string of space-separated words. Each word consists only of lowercase letters.

Example 1:

`Input:`

```
A = "apple banana mango"
B = "banana fruits mango"
```

`Output:`

```
['apple', 'fruits']
```

In [8]:

```
A = "apple banana mango"
```

```
B = "banana fruits mango"

A1=A.split()
B1=B.split()

unique_word=[]

for i in A1:
    if i not in B1:
        unique_word.append(i)

for i in B1:
    if i not in A1:
        unique_word.append(i)

print(unique_word)
```

```
['apple', 'fruits']
```

`Problem 13` : Word location in String.

Statement: Find a location of a word in a given sentence.

Example 1:

`Input:`

```
Sentence: We can learn data science through campusx mentorship program.

word: campusx
```

`Output:`

```
Location of the word is 7.
```

In [13]:

```
Sentence="We can learn data science through campusx mentorship program."
S1=Sentence.split()
print(S1)

S1.index("campusx")
```

```
['We', 'can', 'learn', 'data', 'science', 'through', 'campusx', 'mentorship', 'program.']
```

Out[13]:

```
6
```

In [15]:

```
#OR
position=0
for index,i in enumerate(S1):
    if i=="campusx":
        position=index+1

print(position)
```

```
7
```

`Problem 20` : Write a program that can remove all the duplicate characters from a string. User will provide the input.

In [19]:

```
input_str = input("Enter a string: ")

# Convert string to set to remove duplicates
string_set = set(input_str)

# Join set back to string
no_dup_str = ''.join(str_set)

print("String after removing duplicates: ", no_dup_str)
```

```
Enter a string: apple
String after removing duplicates:  pael
```